

LQG s hyperstavem

4. března 2012

0.1 Systém

Jako systémem uvažujeme PMSM a předpokládáme jeho popis pomocí následujících diskretních rovnic:

$$\begin{aligned}
 i_{\alpha,t+1} &= ai_{\alpha,t} + b\omega_t \sin \vartheta_t + cu_{\alpha,t} \\
 i_{\beta,t+1} &= ai_{\beta,t} - b\omega_t \cos \vartheta_t + cu_{\beta,t} \\
 \omega_{t+1} &= d\omega_t + e(i_{\beta,t} \cos \vartheta_t - i_{\alpha,t} \sin \vartheta_t) \\
 \vartheta_{t+1} &= \vartheta_t + \Delta t \omega_t
 \end{aligned} \tag{0.1}$$

kde $i_{\alpha\beta}$ představují proudy v osách α a β , $u_{\alpha\beta}$ napětí v jednotlivých osách, ω je hodnota otáček a ϑ je poloha (úhel natočení). Konstantní na čase nezávislé parametry a, b, c, d, e předpokládáme známé, Δt je diskretní časový krok a dolní indexy t a $t+1$ představují diskretní čas.

Definujme stav systému v čase t jako

$$x_t = (i_{\alpha,t}, i_{\beta,t}, \omega_t, \vartheta_t)^T$$

dále řízení v čase t jako

$$u_t = (u_{\alpha,t}, u_{\beta,t})^T$$

a výstup (měření) v čase t jako

$$y_t = (y_{\alpha,t}, y_{\beta,t})^T$$

přičemž význam použitých symbolů vychází z rovnic (0.1). Když dále uvažujeme aditivní bílý gaussovský šum, získáme zápis systému ve tvaru

$$\begin{aligned}
 x_{t+1} &= f(x_t, u_t) + v_t \\
 y_t &= h(x_t) + w_t
 \end{aligned} \tag{0.2}$$

kde v_t a w_t představují náhodné veličiny s normálním rozdělením s nulovou střední hodnotou a kovariančními maticemi V a W v tomto pořadí a funkce f a h jsou definovány následovně:

$$\begin{aligned}
 f(x_t, u_t) &= \begin{pmatrix} ai_{\alpha,t} + b\omega_t \sin \vartheta_t + cu_{\alpha,t} \\ ai_{\beta,t} - b\omega_t \cos \vartheta_t + cu_{\beta,t} \\ d\omega_t + e(i_{\beta,t} \cos \vartheta_t - i_{\alpha,t} \sin \vartheta_t) \\ \vartheta_t + \Delta t \omega_t \end{pmatrix} \\
 h(x_t) &= \begin{pmatrix} i_{\alpha,t} \\ i_{\beta,t} \end{pmatrix}
 \end{aligned} \tag{0.3}$$

Redukovaný model

Z úsporných důvodů může být někdy výhodnější namísto popisu systému uvedeného výše (dále budeme označovat jako *plný model*), který vychází z rovnic (0.2) a (0.3) použít jeho redukovanou verzi v následujícím tvaru:

Vektor stavu x_t bude mít jen dvě složky

$$x_t = (\omega_t, \vartheta_t)^T$$

a pro výstup (měření) y_t využijeme toho, že proudy přímo měříme (i když ne zcela přesně)

$$y_t = (i_{\alpha,t}, i_{\beta,t})^T$$

Rovnice systému (0.2) a (0.3) pak zapíšeme ve tvaru

$$\begin{aligned} x_{t+1} &= f(x_t, y_t) + \bar{v}_t \\ y_t &= h(x_t, y_t, u_t) + \bar{w}_t \end{aligned} \quad (0.4)$$

$$\begin{aligned} f(x_t, y_t) &= \begin{pmatrix} d\omega_t + e(i_{b,t} \cos \vartheta_t - i_{\alpha,t} \sin \vartheta_t) \\ \vartheta_t + \Delta t \omega_t \end{pmatrix} \\ h(x_t, y_t, u_t) &= \begin{pmatrix} ai_{\alpha,t} + b\omega_t \sin \vartheta_t + cu_{\alpha,t} \\ ai_{\beta,t} - b\omega_t \cos \vartheta_t + cu_{\beta,t} \end{pmatrix} \end{aligned} \quad (0.5)$$

dále je pak třeba upravit kovarianční matice šumu V a W . Matici V je nutno předpokládat v blokově diagonálním tvaru

$$V = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}$$

a jako nové kovarianční matice označíme $\bar{V} = V_2$ a $\bar{W} = V_1 + W$.

0.2 EKF

Rozšířený Kalmanův filter (EKF) nahrazuje skutečný nelineární systém (0.2 a 0.3 případně 0.4 a 0.5 pro redukovaný model) lineárním

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + \tilde{v}_t \\ y_t &= C_t x_t + \tilde{w}_t \end{aligned} \quad (0.6)$$

kde do šumů \tilde{v}_t a \tilde{w}_t je možno zahrnout nepřesnosti linearizace tím, že se zvětší jejich kovariance oproti původním v_t a w_t respektive \bar{v}_t a \bar{w}_t v případě redukovaného modelu. Ostatní označení odpovídá nelineárním rovnicím PMSM (0.2 a 0.3 případně 0.4 a 0.5) s tím, že matice A , B a C vzniknou linearizací jako

$$\begin{aligned} A_t &= \frac{\partial f(x_t, u_t)}{\partial x_t} \\ B_t &= \frac{\partial f(x_t, u_t)}{\partial u_t} \\ C_t &= \frac{\partial h(x_t)}{\partial x_t} \end{aligned}$$

Matice derivací

Konkrétně pro PMSM s funkcemi f a h danými vztahem (0.3) jsou matice A , B a C následující:

$$\begin{aligned}
 A_t &= \begin{bmatrix} a & 0 & b \sin \vartheta_t & b\omega_t \cos \vartheta_t \\ 0 & a & -b \cos \vartheta_t & b\omega_t \sin \vartheta_t \\ -e \sin \vartheta_t & e \cos \vartheta_t & d & -e(i_{\beta,t} \sin \vartheta_t + i_{\alpha,t} \cos \vartheta_t) \\ 0 & 0 & \Delta t & 1 \end{bmatrix} \\
 B &= \begin{bmatrix} c & 0 \\ 0 & c \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{0.7}$$

Pro redukovaný model jsou matice A a C ve tvaru

$$\begin{aligned}
 A_t &= \begin{bmatrix} d & -e(i_{\beta,t} \sin \vartheta_t + i_{\alpha,t} \cos \vartheta_t) \\ \Delta t & 1 \end{bmatrix} \\
 C_t &= \begin{bmatrix} b \sin \vartheta_t & b\omega_t \cos \vartheta_t \\ -b \cos \vartheta_t & b\omega_t \sin \vartheta_t \end{bmatrix}
 \end{aligned} \tag{0.8}$$

Matice B pro redukovaný model uvedena není, protože pro samotný výpočet EKF není třeba a problematika lineárně kvadratického řízení pro redukovaný model bude rozebrána dále, viz část (0.3).

Rovnice EKF

Následně lze užít algoritmu formálně shodného s klasickým Kalmanovým filtrem, kde místo lineárního systému je užít systém linearizovaný:

predikce (time update)

$$\begin{aligned}
 \bar{\hat{x}}_t &= f(\hat{x}_{t-1}, u_{t-1}) \\
 \bar{P}_t &= A_{t-1} P_{t-1} A_{t-1}^T + V_{t-1}
 \end{aligned} \tag{0.9}$$

korekce (data update)

$$\begin{aligned}
 S_t &= C_t \bar{P}_t C_t^T + W_t \\
 K_t &= \bar{P}_t C_t^T S_t^{-1} \\
 P_t &= (I - K_t C_t) \bar{P}_t \\
 \hat{y}_t &= y_t - h(\bar{\hat{x}}_t) \\
 \hat{x}_t &= \bar{\hat{x}}_t + K_t \hat{y}_t
 \end{aligned} \tag{0.10}$$

0.3 Lineárně kvadratické řízení

Tento algoritmus opět předpokládá lineární systém, kterým PMSM není. Chceme opět získat systém ve tvaru (0.6) a je tedy nutné provést linearizaci. Nelze ale přímo použít matice získané v předchozí části (0.2) zabývající se EKF. Zde je nutné vycházet z Taylo-rova rozvoje a zohlednit i případné konstantní členy. Obecně pro funkci $f(x)$ má rozvoj do prvního řádu v nějakém bodě x_0 tvar

$$f(x) \cong f(x_0) + \frac{\partial f}{\partial x}(x_0)(x - x_0)$$

kde parciální derivací f dle x je matice A z předchozí části (0.2) o EKF vypočtená v bodě x_0 a tedy

$$f(x) \cong Ax + (f(x_0) - Ax_0) = Ax + \gamma$$

kde vektor γ představuje konstantní člen (nezávisí na x) a předchozí rovnice tedy není homogenní, jak bychom potřebovali jako výsledek linearizace (0.6). Proto tedy zvětšíme matici A o 1 (o jeden sloupec a řádek) a stejně tak zvětšíme i stav o 1 (přidáme konstantu) a předchozí rovnici získáme ve tvaru

$$\begin{pmatrix} f(x) \\ 1 \end{pmatrix} \cong \bar{A} \begin{pmatrix} x \\ 1 \end{pmatrix}$$

kde

$$\bar{A} = \begin{bmatrix} A & (f(x_0) - Ax_0) \\ 0 & 1 \end{bmatrix}$$

přičemž 0 zde označuje nulový řádkový vektor vhodné velikosti. Tímto postupem lze již získat požadovaný lineární popis (0.6), který současně zohledňuje i konstantní členy.

Matice pro LQ

Pro případ plného stavu je matice A_t dána vztahem (0.7), kde jako hodnoty stavových veličin (složek vektoru x_t) použijeme hodnoty bodu x_0 , ve kterém linearizujeme. Konstantní člen $\gamma = f(x_0) - A_t x_0$ tedy vypočteme jako

$$\gamma = \begin{pmatrix} -b\omega_0\vartheta_0 \cos \vartheta_0 \\ -b\omega_0\vartheta_0 \sin \vartheta_0 \\ e\vartheta_0 (i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) \\ 0 \end{pmatrix}$$

kde dolní index 0 neznačí nulový čas, ale bod linearizace x_0 . Matice \bar{A}_t vypočtená v bodě x_0 (složky x_0 budou opět značeny dolním indexem 0) pak je

$$\begin{bmatrix} a & 0 & b \sin \vartheta_0 & b\omega_0 \cos \vartheta_0 & -b\omega_0\vartheta_0 \cos \vartheta_0 \\ 0 & a & -b \cos \vartheta_0 & b\omega_0 \sin \vartheta_0 & -b\omega_0\vartheta_0 \sin \vartheta_0 \\ -e \sin \vartheta_0 & e \cos \vartheta_0 & d & -e (i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) & e\vartheta_0 (i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) \\ 0 & 0 & \Delta t & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Matici B_t derivací $f(x_t, u_t)$ dle vstupů u_t lze volit konstantní a časově nezávislou ve tvaru (0.7), protože funkce f je ve vstupech u lineární.

Ztrátová funkce

Protože chceme využít lineárně kvadratického algoritmu, je třeba formulovat ztrátovou funkci jako aditivní a kvadratickou, obecně ve tvaru

$$\mathbf{E} \left\{ x_T^T Q_T x_T + \sum_{t=0}^{T-1} (x_t^T Q_t x_t + u_t^T R_t u_t) \right\} \quad (0.11)$$

kde \mathbf{E} značí očekávanou hodnotu, Q_t a R_t jsou penalizační matice stavu systému (splnění požadavků řízení) a penalizace vstupů.

Hlavním požadavkem na systém je dosažení požadované hodnoty otáček $\bar{\omega}_t$ v čase t . Výše navržená ztráta (0.11) však vede na řízení pouze na nulovou hodnotu odpovídající $\bar{\omega} \equiv 0$, pro řízení na nenulové požadované otáčky je třeba modifikovat stav systému a zavést substituci

$$\psi_t = \omega_t - \bar{\omega}_t$$

a veličinu ψ pak již řídíme na nulovou hodnotu. Tuto substituci, která závisí na parametru $\bar{\omega}$, je třeba zanést do všech rovnic. Ve stavu systému veličina ψ_t nahradí veličinu ω_t . Dále je třeba zahrnout i všechny konstantní členy, které v důsledku substituce vzniknou.

Penalizační matici stavu systému v (0.11) budeme uvažovat nezávislou na čase $Q_t = Q$ pro všechna t a ve tvaru

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (0.12)$$

kde q je pevně zvolená konstanta a matice Q má již rozměr 5×5 , protože byl stav rozšířen o konstantní člen v důsledku linearizace. Konečnou matici Q_T budeme uvažovat nulovou.

Dalším požadavkem je omezení na napětí – vstupy do systému, vyjádřené pomocí maximálního napětí U_{max} , které je schopen poskytnout napájecí zdroj. Toto omezení můžeme zapsat jako

$$\|u_t\| \leq U_{max} \quad (0.13)$$

případně jako omezení na každou složku vektoru u_t zvlášť. Tento požadavek nelze přímo zapsat jako kvadratickou funkci a proto je třeba vhodně zvolit matici R_t v (0.11) aby dostatečně penalizovala příliš velké hodnoty řízení u_t a dále počítat s tím, že při přesažení hodnoty U_{max} dojde k ořezu. Penalizační matici řízení opět volíme nezávislou na čase, tj. $R_t = R$ pro všechna t , a ve tvaru

$$R = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}$$

kde r je zvolená konstanta.

Substituované rovnice

V důsledku substituce $\psi_t = \omega_t - \bar{\omega}_t$ se rovnice (0.1) změni na

$$\begin{aligned} i_{\alpha,t+1} &= ai_{\alpha,t} + b(\psi_t + \bar{\omega}_t) \sin \vartheta_t + cu_{\alpha,t} \\ i_{\beta,t+1} &= ai_{\beta,t} - b(\psi_t + \bar{\omega}_t) \cos \vartheta_t + cu_{\beta,t} \\ \psi_{t+1} &= d\psi_t + e(i_{\beta,t} \cos \vartheta_t - i_{\alpha,t} \sin \vartheta_t) \\ \vartheta_{t+1} &= \vartheta_t + \Delta t (\psi_t + \bar{\omega}_t) \end{aligned} \quad (0.14)$$

předpokládáme-li, že pro požadované otáčky $\bar{\omega}$ platí $\bar{\omega}_{t+1} = d\bar{\omega}_t$.

Derivováním těchto rovnic dle nového stavu (substituovaného) $(i_{\alpha,t}, i_{\beta,t}, \psi_t, \vartheta_t)^T$ získáme matici

$$A_t = \begin{bmatrix} a & 0 & b \sin \vartheta_t & b(\psi_t + \bar{\omega}_t) \cos \vartheta_t \\ 0 & a & -b \cos \vartheta_t & b(\psi_t + \bar{\omega}_t) \sin \vartheta_t \\ -e \sin \vartheta_t & e \cos \vartheta_t & d & -e(i_{\beta,t} \sin \vartheta_t + i_{\alpha,t} \cos \vartheta_t) \\ 0 & 0 & \Delta t & 1 \end{bmatrix}$$

kteřá je hodnotově stejná s maticí A_t získanou na základě původního nesubstituovaného stavu (tj. s $x^{(3)} = \omega$).

Konstantní člen $\gamma = f(x_0) - A_t x_0$ je však již jiný a závisí na hodnotě $\bar{\omega}_t$, která do něj vstupuje jako časově proměnný parametr.

$$\gamma_{\bar{\omega}_t} = \begin{pmatrix} -b\omega_0 \vartheta_0 \cos \vartheta_0 + b\bar{\omega}_t \sin \vartheta_0 \\ -b\omega_0 \vartheta_0 \sin \vartheta_0 - b\bar{\omega}_t \cos \vartheta_0 \\ e\vartheta_0 (i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) \\ \Delta t \bar{\omega}_t \end{pmatrix}$$

Výsledná matice \bar{A}_t je pak ve tvaru

$$\begin{bmatrix} a & 0 & b \sin \vartheta_0 & b\omega_0 \cos \vartheta_0 & -b\omega_0 \vartheta_0 \cos \vartheta_0 + b\bar{\omega}_t \sin \vartheta_0 \\ 0 & a & -b \cos \vartheta_0 & b\omega_0 \sin \vartheta_0 & -b\omega_0 \vartheta_0 \sin \vartheta_0 - b\bar{\omega}_t \cos \vartheta_0 \\ -e \sin \vartheta_0 & e \cos \vartheta_0 & d & -e(i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) & e\vartheta_0 (i_{\beta,0} \sin \vartheta_0 + i_{\alpha,0} \cos \vartheta_0) \\ 0 & 0 & \Delta t & 1 & \Delta t \bar{\omega}_t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Bellmanova funkce

Cílem úlohy je minimalizovat ztrátovou funkci (0.11). Klasickým postupem pro řešení této úlohy je užití Bellmanovy funkce a algoritmu dynamického programování:

V koncovém čase T položíme

$$V_T(x_T) = 0 \quad (0.15)$$

a dále počítáme zpět v čase

$$V_{t-1}(x_{t-1}, u_{t-1}) = \min_{u_{t-1}} E \{x_t^T Q_t x_t + u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t\} \quad (0.16)$$

pro t od $T - 1$ do 1, kde střední hodnota je podmíněna \mathcal{I}_t , které reprezentuje současně dostupnou informaci o systému zahrnující všechna měření a řídicí vstupy do času t .

Uvažovanou kvadratickou ztrátu za jeden časový krok

$$x_t^T Q_t x_t + u_t^T R_t u_t$$

při konkrétní volbě matice Q ve tvaru (0.12) přejde na

$$q \left(x_t^{(3)} - \bar{\omega}_t \right) + u_t^T R_t u_t$$

kde horní index v závorce značí složku vektoru. Pak je možno rovnici (0.16) dále zjednodušit

$$\begin{aligned} V_{t-1}(x_{t-1}, u_{t-1}) &= \min_{u_{t-1}} \mathbb{E} \left\{ x_t^T Q_t x_t + u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \\ &= \min_{u_{t-1}} \left(\mathbb{E} \left\{ q \left(x_t^{(3)} - \bar{\omega}_t \right) \right\} + \mathbb{E} \left\{ u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \right) \\ &= \min_{u_{t-1}} \left(q \mathbb{E} \left\{ \left(x_t^{(3)} \right)^2 + \bar{\omega}_t^2 + 2x_t^{(3)} \bar{\omega}_t \right\} + \mathbb{E} \left\{ u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \right) \quad (0.17) \\ &= \min_{u_{t-1}} \left(q \left(\mathbb{E} \left\{ \left(x_t^{(3)} \right)^2 \right\} + \mathbb{E} \left\{ \bar{\omega}_t^2 \right\} + \mathbb{E} \left\{ 2x_t^{(3)} \bar{\omega}_t \right\} \right) + \mathbb{E} \left\{ u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \right) \\ &= \min_{u_{t-1}} \left(q \left(\left(\hat{x}_t^{(3)} \right)^2 + \text{Var} \left(x_t^{(3)} \right) + \bar{\omega}_t^2 + 2\hat{x}_t^{(3)} \bar{\omega}_t \right) + \mathbb{E} \left\{ u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \right) \\ &= \min_{u_{t-1}} \left(q \left(\hat{x}_t^{(3)} - \bar{\omega}_t \right) + q \text{Var} \left(x_t^{(3)} \right) + \mathbb{E} \left\{ u_t^T R_t u_t + V_t(x_t, u_t) \mid \mathcal{I}_t \right\} \right) \end{aligned}$$

kde \hat{x} označuje střední hodnotu x a dále jsme využili toho, že $\bar{\omega}_t$ je daný parametr a tedy je pro výpočet střední hodnoty konstantou a vztahu $\text{Var}(x) = \mathbb{E}\{x^2\} - (\mathbb{E}\{x\})^2$. Tedy ve výpočtu Bellmanovy funkce V v rovnici (0.16) můžeme náhodnou veličinu x_t nahradit její střední hodnotou \hat{x}_t , když navíc zahrneme do rovnice varianci třetí složky x_t , tj. varianci otáček stroje.

Výpočet lineárně kvadratického řízení

Pro samotný výpočet lineárně kvadratického řízení je užito následujících rovnic

$$\begin{aligned} K_T &= Q_T \\ K_t &= A_t^T \left(K_{t+1} - K_{t+1} B_t (B_t^T K_{t+1} B_t + R_t)^{-1} B_t^T K_{t+1} \right) A_t + Q_t \\ L_t &= - (B_t^T K_{t+1} B_t + R_t)^{-1} B_t^T K_{t+1} A_t \\ u_t &= L_t x_t \end{aligned}$$

Tyto rovnice by měly být napočítávány v čase zpět (od koncového času) až do aktuálního času. Protože ale systém vznikl linearizací v nějakém reprezentativním bodě, který se s

vývojem systému mění, je třeba celý výpočet znovu provést v každém časovém kroku. Proto je výhodnější si výpočet usnadnit například využitím ubíhajícího horizontu.

Při výpočtu řízení u_t pomocí matice L_t je třeba dosadit za vektor x_t správné hodnoty, konkrétně v důsledku nenulové požadované hodnoty $\bar{\omega}$ za třetí složku vektoru x_t není dosazena hodnota ω_t , ale substituovaná $\psi_t = \omega_t - \bar{\omega}_t$.

Předchozí výpočet pomocí Riccatiho rovnice však není příliš vhodným z numerických důvodů (**nějaká reference**). Místo něj pro praktické výpočty použijeme algoritmus lineárně kvadratického řízení založený na QR rozkladu (**reference**). Tento algoritmus má lepší numerické vlastnosti, umožňuje snadnější výpočet maticové inverze (inverze pouze trojúhelníkové matice) a lze pomocí něj implementovat i složitější kvadratickou ztrátovou funkci (nejen dva členy pro penalizaci stavu a vstupů).

Postup je založen na přepisu kvadratické ztráty do tvaru

$$x_{t+1}^T Q_t x_{t+1} + u_t^T R_t u_t = x_{t+1}^T \sqrt{Q_t}^T \sqrt{Q_t} x_{t+1} + u_t^T \sqrt{R_t}^T \sqrt{R_t} u_t$$

kde $\sqrt{\cdot}$ je vhodná maticová odmocnina. A tedy v každém časovém kroku t minimalizujeme funkci

$$x_{t+1}^T \sqrt{Q_t}^T \sqrt{Q_t} x_{t+1} + u_t^T \sqrt{R_t}^T \sqrt{R_t} u_t + x_{t+1}^T \sqrt{S_t}^T \sqrt{S_t} x_{t+1}$$

kde S_t reprezentuje ztrátu v následujících časových krocích až do konce časového horizontu. Do tohoto kvadratického výrazu je možno dostadit model vývoje pro $x_{t+1} = A_t x_t + B_t u_t$

$$(A_t x_t + B_t u_t)^T \sqrt{Q_t} \sqrt{Q_t} (A_t x_t + B_t u_t) + u_t^T \sqrt{R_t}^T \sqrt{R_t} u_t + (A_t x_t + B_t u_t)^T \sqrt{S_t}^T \sqrt{S_t} (A_t x_t + B_t u_t)$$

a následně jej zapsat maticově ve tvaru

$$\begin{pmatrix} u_t \\ x_t \end{pmatrix}^T \begin{bmatrix} \sqrt{Q_t} B_t & \sqrt{Q_t} A_t \\ \sqrt{R_t} & 0 \\ \sqrt{S_t} B_t & \sqrt{S_t} A_t \end{bmatrix}^T \underbrace{\begin{bmatrix} \sqrt{Q_t} B_t & \sqrt{Q_t} A_t \\ \sqrt{R_t} & 0 \\ \sqrt{S_t} B_t & \sqrt{S_t} A_t \end{bmatrix}}_Z \begin{pmatrix} u_t \\ x_t \end{pmatrix}$$

na matici Z následně aplikujeme QR rozklad, to jest $Z = Q_Z R_Z$ a předchozí vztah upravíme na tvar

$$\begin{pmatrix} u_t \\ x_t \end{pmatrix}^T Z^T Z \begin{pmatrix} u_t \\ x_t \end{pmatrix} = \begin{pmatrix} u_t \\ x_t \end{pmatrix}^T R_Z^T Q_Z^T Q_Z R_Z \begin{pmatrix} u_t \\ x_t \end{pmatrix} = \begin{pmatrix} u_t \\ x_t \end{pmatrix}^T R_Z^T R_Z \begin{pmatrix} u_t \\ x_t \end{pmatrix}$$

Matice R_Z je v horním trojúhelníkovém tvaru, tedy blokově zapsáno

$$R_Z = \begin{bmatrix} R_{uu} & R_{ux} \\ 0 & R_{xx} \end{bmatrix}$$

Ztrátu nyní můžeme zapsat jako

$$\begin{aligned} \begin{pmatrix} u_t \\ x_t \end{pmatrix}^T R_Z^T R_Z \begin{pmatrix} u_t \\ x_t \end{pmatrix} &= \begin{pmatrix} R_{uu} u_t + R_{ux} x_t \\ R_{xx} x_t \end{pmatrix}^T \begin{pmatrix} R_{uu} u_t + R_{ux} x_t \\ R_{xx} x_t \end{pmatrix} \\ &= (R_{uu} u_t + R_{ux} x_t)^T (R_{uu} u_t + R_{ux} x_t) + x_t^T R_{xx}^T R_{xx} x_t \end{aligned}$$

kteřou, vzhledem k její kvadratickosti a nezávislosti druhého členu na u_t , zřejmě minimalizujeme volbou u_t takovou, že $(R_{uu}u_t + R_{ux}x_t) = 0$ a tedy volíme

$$u_t = -R_{uu}^{-1}R_{ux}x_t$$

Matici $R_{xx}^T R_{xx}$ pak použijeme do předchozího časového kroku jako novou matici S .

LQ řízení pro redukováný model

Pro redukováný systém samozřejmě platí vše uvedené v předchozím odstavci, řízení je ale komplikovanější, protože ve funkci popisující vývoj systému explicitně nevystupuje řízení u_t . Je tedy třeba vhodným způsobem tento problém vyřešit. Jednou z možností je zřetězení dvou LQ regulátory. V prvním kroku považovat za řízení proudy $i_{\alpha,\beta}$ a tedy tento první regulátor by na výstupu generoval požadované proudy $\bar{i}_{\alpha,\beta}$. Druhý regulátor by pak na základě rovnic pro vývoj proudů a referenčních hodnot proudů $\bar{i}_{\alpha,\beta}$ našel řízení $u_{\alpha,\beta}$.

Matice pro redukováný model

Protože ve funkci $f(x_t, y_t)$ v rovnicích (0.4) a (0.5) explicitně nevystupuje řízení u_t , je třeba zvolit trochu odlišný přístup, než pro plný model. Řízení budeme navrhovat ve dvou krocích. V prvním kroku budeme předpokládat, že vstupem jsou proudy $i_{\alpha\beta}$ a lineárně kvadratický algoritmus bude na svém výstupu produkovat požadované hodnoty těchto proudů $\bar{i}_{\alpha\beta}$. V dalším kroku druhý lineárně kvadratický algoritmus na základě požadovaných proudů $\bar{i}_{\alpha\beta}$ již navrhne hodnotu napětí $u_{\alpha\beta}$.

Dále provedeme ještě drobné zjednodušení a funkci $f(x_t, y_t)$ rozdělíme na dvě části

$$f(x_t, y_t) = \begin{pmatrix} d\omega_t \\ \vartheta_t + \Delta t\omega_t \end{pmatrix} + \begin{pmatrix} e(i_{b,t} \cos \vartheta_t - i_{\alpha,t} \sin \vartheta_t) \\ 0 \end{pmatrix}$$

Matici A_t pak položíme rovnou první matici první, lineární, části systému

$$A = \begin{bmatrix} d & 0 \\ \Delta t & 1 \end{bmatrix}$$

a matici B_t pak získáme linearizací druhé části jako

$$B_t = \begin{bmatrix} -e \sin \vartheta_t & e \cos \vartheta_t \\ 0 & 0 \end{bmatrix}$$

Tento postup neodpovídá přesně postupu odvození derivací užitému pro plný stav. Jeho výhodou však je, že již není třeba přidávat konstantní členy jako důsledek linearizace. Snadněji se také zahrne požadavek na nenulovou referenční hodnotu $\bar{\omega}$. Následně je užito lineárně kvadratického algoritmu s výše popsánými maticemi.

Ve druhém kroku pak na základě referenčních hodnot proudů $\bar{i}_{\alpha\beta}$ nalezneme požadované řízení $u_{\alpha\beta}$. Využijeme k tomu rovnic pro funkci $h(x_t, y_t, u_t)$ viz (0.5)

$$h(x_t, y_t, u_t) = \begin{pmatrix} ai_{\alpha,t} + b\omega_t \sin \vartheta_t + cu_{\alpha,t} \\ ai_{\beta,t} - b\omega_t \cos \vartheta_t + cu_{\beta,t} \end{pmatrix}$$

kteřé jsou v proudech $i_{\alpha\beta}$ i napětích $u_{\alpha\beta}$ lineární a lze opět použít lineárně kvadratický algoritmus. Členy $\pm b\omega_t \frac{\sin}{\cos} \vartheta_t$ zde pak vystupují jako konstanty a projeví se jako korekce vynásobená konstantou $\frac{1}{c}$ odečtená od výsledku.

0.4 LQG s hyperstavem

Následující postup s hyperstavem vychází s článku (Kim2006) [Stochastic Feedback Controller Design Considering the Dual Effect, Kim J., Rock S. M., 2006]. V tomto článku je však narozdíl od následujícího postupu používán spojitý čas.

Jedná se o analogii s LQG v předchozí části, s tím rozdílem, že použijem EKF algoritmus v jistém smyslu jakoby dvakrát. Protože tímto přístupem již značně narůstá dimenzionalita úlohy je z výpočetních důvodů výhodnější užití redukováného modelu, i přes komplikace, které způsobuje při řízení.

Hyperstav

Vyjdeme z redukováného stavu

$$x_t = (\omega_t, \vartheta_t)^T$$

a na něj formálně aplikujeme EKF. Tím získáme, kromě odhadu stavu x_t i odhad jeho variance v podobě matice

$$P = \begin{bmatrix} P_\omega & P_{\omega\vartheta} \\ P_{\omega\vartheta} & P_\vartheta \end{bmatrix}$$

a současně rovnice EKF (0.9) a (0.10) představují předpis pro výpočet P :

$$\begin{aligned} \bar{P} &= APA^T + V \\ S &= C\bar{P}C^T + W \\ K &= \bar{P}C^T S^{-1} \\ P^+ &= (I - KC)\bar{P} \end{aligned} \tag{0.18}$$

kde jsou z důvodu jednoduššího zápisu vynechány časové indexy t a místo nic je užit horní index $+$ pro hodnotu v následujícím čase $t + 1$.

Nyní definujeme *hyperstav* ξ_t v čase t jako

$$\xi_t = (\omega_t, \vartheta_t, P_\omega, P_{\omega\vartheta}, P_\vartheta)^T$$

Na hyperstav již aplikujeme algoritmus pro LQG, jak byl popsán v předchozí části. Problém však představuje nalezení matice derivací A , protože je třeba derivovat maticové rovnice pro výpočet EKF (0.18) pro stavu x . Jedním ze způsobů jak je to možné provést

je derivovat každou z rovnic (0.18) dle jednotlivých složek vektoru ξ :

$$\begin{aligned}
\frac{\partial \bar{P}}{\partial \xi_i} &= \frac{\partial A}{\partial \xi_i} P A^T + A \frac{\partial P}{\partial \xi_i} A^T + A P \frac{\partial A^T}{\partial \xi_i} \\
\frac{\partial S}{\partial \xi_i} &= \frac{\partial C}{\partial \xi_i} \bar{P} C^T + C \frac{\partial \bar{P}}{\partial \xi_i} C^T + C \bar{P} \frac{\partial C^T}{\partial \xi_i} \\
\frac{\partial K}{\partial \xi_i} &= \frac{\partial \bar{P}}{\partial \xi_i} C^T S^{-1} + \bar{P} \frac{\partial C^T}{\partial \xi_i} S^{-1} - \bar{P} C^T S^{-1} \frac{\partial S}{\partial \xi_i} S^{-1} \\
\frac{\partial P^+}{\partial \xi_i} &= \frac{\partial \bar{P}}{\partial \xi_i} - \frac{\partial K}{\partial \xi_i} C \bar{P} - K \frac{\partial C}{\partial \xi_i} \bar{P} - K C \frac{\partial \bar{P}}{\partial \xi_i}
\end{aligned} \tag{0.19}$$

kde $\frac{\partial}{\partial \xi_i}$ představuje zápis derivace dle i -té složky vektoru ξ a matice V a W uvažujeme jako konstanty v ξ . Matice linearizovaného vývoje hyperstavu A_{hyp} bude mít nyní blokový tvar

$$A_{hyp} = \begin{bmatrix} A_1 & A_2 & 0 & 0 & 0 \\ \left(\frac{\partial P^+}{\partial \omega}\right)^{sl} & \left(\frac{\partial P^+}{\partial \vartheta}\right)^{sl} & \left(\frac{\partial P^+}{\partial P_\omega}\right)^{sl} & \left(\frac{\partial P^+}{\partial P_{\omega\vartheta}}\right)^{sl} & \left(\frac{\partial P^+}{\partial P_\vartheta}\right)^{sl} \end{bmatrix}$$

kde A_i představuje i -tý sloupec matice A , zápis 0 je sloupec nul vhodné délky a parciální derivace P^+ dle složky ξ_i v závorce s horním indexem sl $\left(\frac{\partial P^+}{\partial \xi_i}\right)^{sl}$ je myšlena v tom smyslu, že po vypočtení příslušné derivace $\frac{\partial P^+}{\partial \xi_i}$ z rovnice (0.19) jsou z této matice vybrány 3 z jejích 4 prvků tvořící horní nebo dolní trojúhelník a zapísány je ve smyslu tvorby vektoru hyperstavu do sloupce:

$$\begin{aligned}
\frac{\partial P^+}{\partial \xi_i} &= \begin{bmatrix} \frac{\partial P_\omega^+}{\partial \xi_i} & \frac{\partial P_{\omega\vartheta}^+}{\partial \xi_i} \\ \frac{\partial P_{\omega\vartheta}^+}{\partial \xi_i} & \frac{\partial P_\vartheta^+}{\partial \xi_i} \end{bmatrix} \\
\left(\frac{\partial P^+}{\partial \xi_i}\right)^{sl} &= \left(\frac{\partial P_\omega^+}{\partial \xi_i} \quad \frac{\partial P_{\omega\vartheta}^+}{\partial \xi_i} \quad \frac{\partial P_\vartheta^+}{\partial \xi_i} \right)^T
\end{aligned}$$

Matici A_{hyp} vzniklou předchozím postupem již můžeme použít v algoritmu EKF pro hyperstav. Jako matici pozorování C_{hyp} použijeme původní matici C pouze doplněnou nulami na vhodný rozměr. Pro lineárně kvadratické řízení platí opět totéž, co pro jednoduché (tj. bez hyperstavu) a matici A_{hyp} je tedy třeba rozšířit zahrnutím konstantních členů, dále je třeba ošetřit substitucí řízení na nenulové požadované otáčky $\bar{\omega}$.

Protože uvažujeme redukovaný model je třeba užít zřetězení dvou LQ regulátorů. Výhodou využití hyperstavu ale je, že máme k dispozici i odhady variancí P původního stavu a tedy je možno zahrnout do kritéria například penalizaci P_ω , která vystupuje v Bellmanově funkci viz vzorec (0.17).

Plný model

Analogicky lze postupovat i pro plný model, všechny odpovídající matice však budou podstatně větší, protože velikost hyperstavu narůstá řádově kvadraticky.

Tedy pro stav

$$x_t = (i_{\alpha,t}, i_{\beta,t}, \omega_t, \vartheta_t)^T$$

vypočteme z EKF kovarianční matici

$$P = \begin{bmatrix} P_5 & P_6 & P_8 & P_{11} \\ P_6 & P_7 & P_9 & P_{12} \\ P_8 & P_9 & P_{10} & P_{13} \\ P_{11} & P_{12} & P_{13} & P_{14} \end{bmatrix}$$

a definujeme *hyperstav* ξ_t v čase t jako

$$\xi_t = (i_{\alpha,t}, i_{\beta,t}, \omega_t, \vartheta_t, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14})^T$$

Rovnice pro výpočet matice P , a tedy i jejích prvků P_i , jsou formálně shodné s rovnicemi pro redukovaný model, pouze rozměry vystupujících matic jsou větší. A matice A_{hyp} je ve tvaru

$$A_{hyp} = \begin{bmatrix} A & 0 \\ \left(\frac{\partial P^+}{\partial P_i} \right)_{i \in \{1 \dots 14\}}^{sl} \end{bmatrix}$$

0.5 Experimenty

Použité nastavení experimentů

Pro simulování chování PMSM byly použity dva typy simulátorů. Prvním byla pouze jednoduchá implementace rovnic popisujících PMSM. Druhou testovanou možností bylo využití simulátoru PMSM (**reference**).

Testování probíhalo na horizontu 120000 časových vzorků, což odpovídá 15s. Ve všech případech byly užity odhadovací a řídicí algoritmy předpokládající stejnou indukčnost v osách d a q . (*pro různé indukčnosti jsou složitější rovnice, tedy mnohem složitější matice derivací a velmi těžko se napočítávají kompenzace v důsledku konstantních členů*). Testování probíhalo na různých profilech požadovaných (referenčních) otáček:

(0) nulové požadované otáčky pro všechna t

(± 1) trojúhelníkové pulzy v rozmezí ± 1

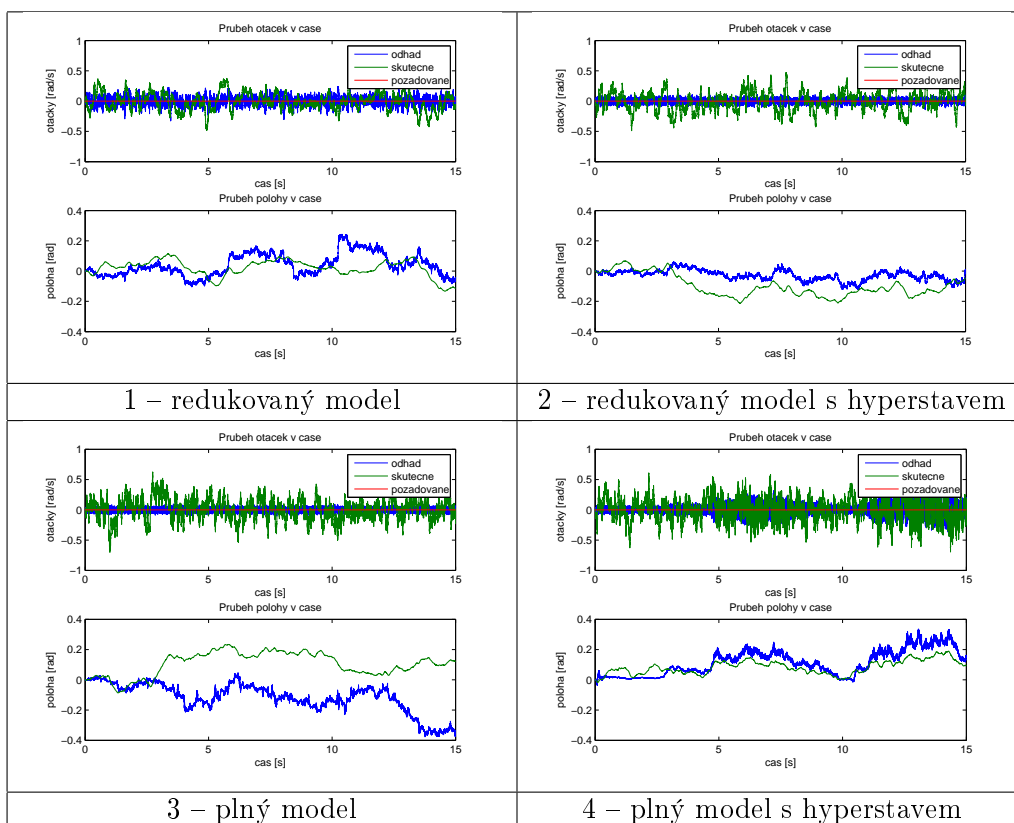
(± 10) trojúhelníkové pulzy v rozmezí ± 10

(± 200) trojúhelníkové pulzy v rozmezí ± 200

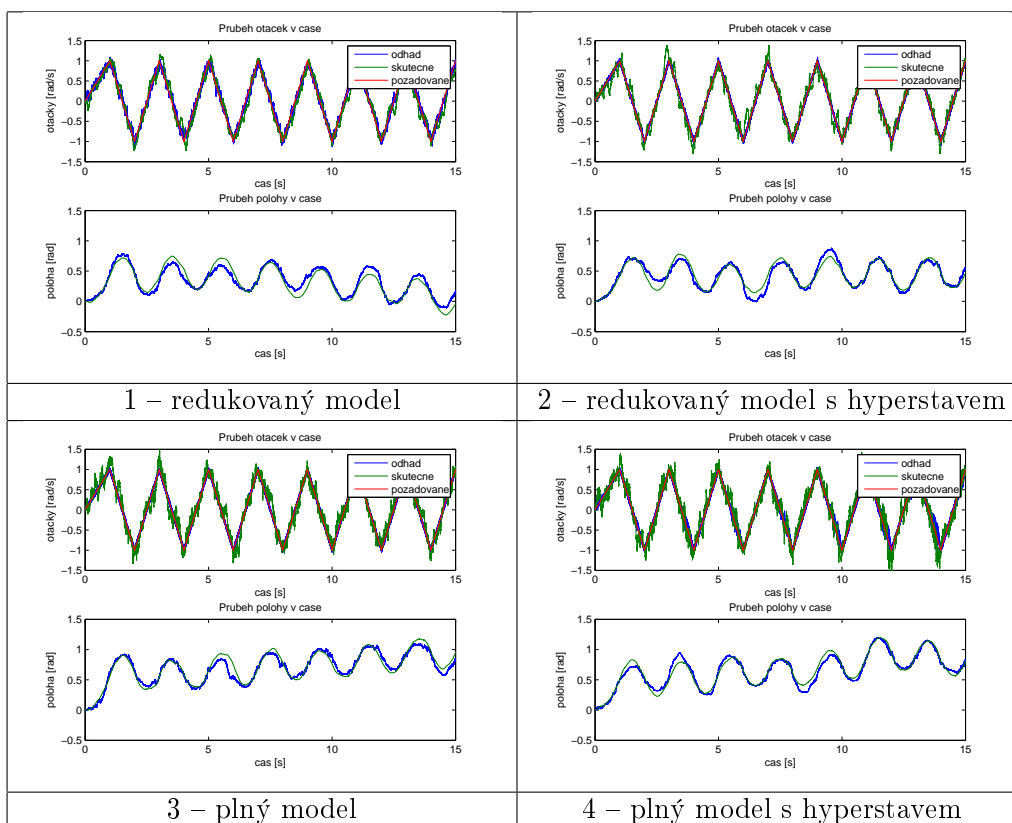
Dále pak byl testován i vliv špatného počátečního odhadu polohy (úhlu natočení) ϑ_0 .

Při použití simulátoru PMSM se velmi často vyskytovaly nedostatky způsobené úbytky napětí, proto byla testována i verze upraveného simulátoru, která se snažila úbytky kompenzovat.

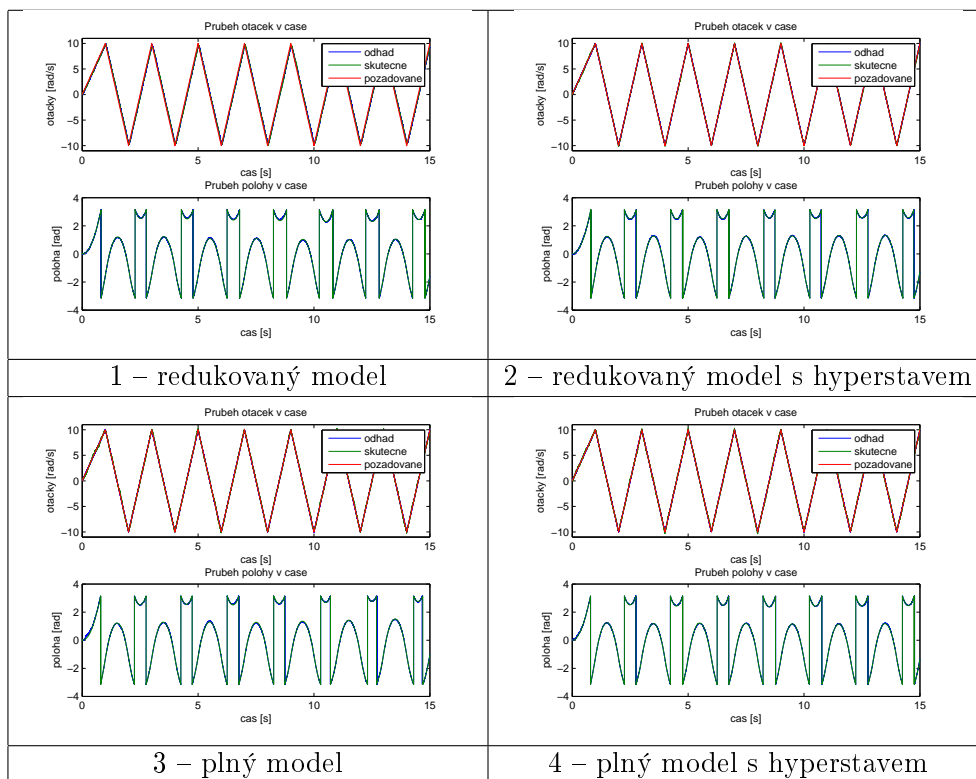
Testovány byly celkem čtyři různé modely: redukovaný model s hyperstavem i bez něj a plný model s hyperstavem a bez něj.



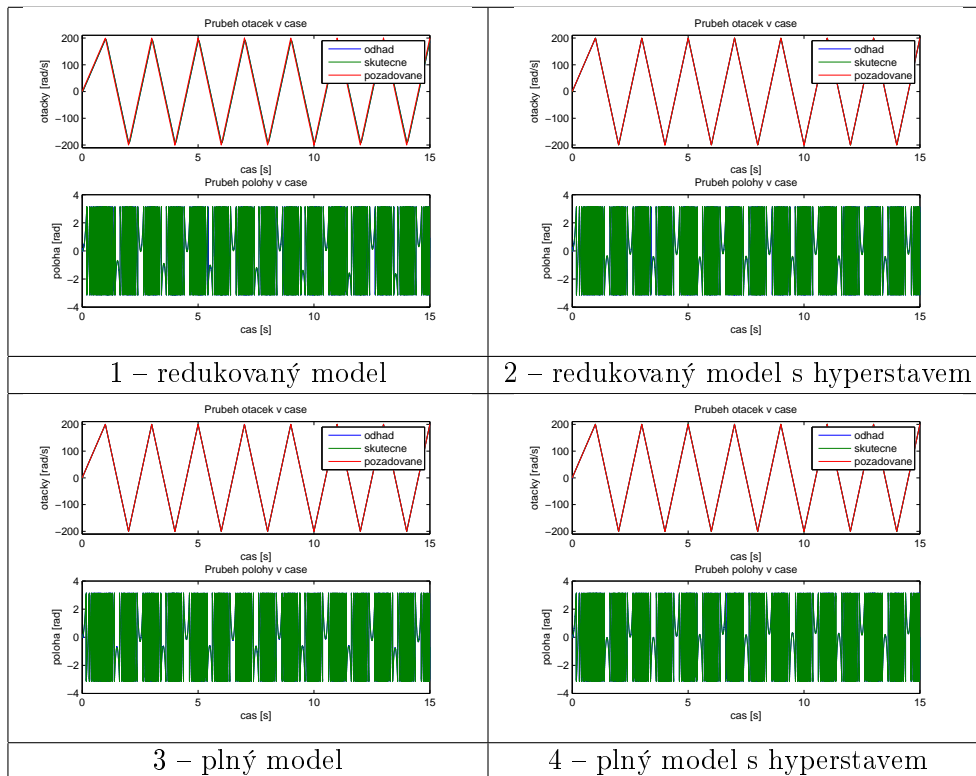
Obrázek 0.1: Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru s nulovými požadovanými otáčkami



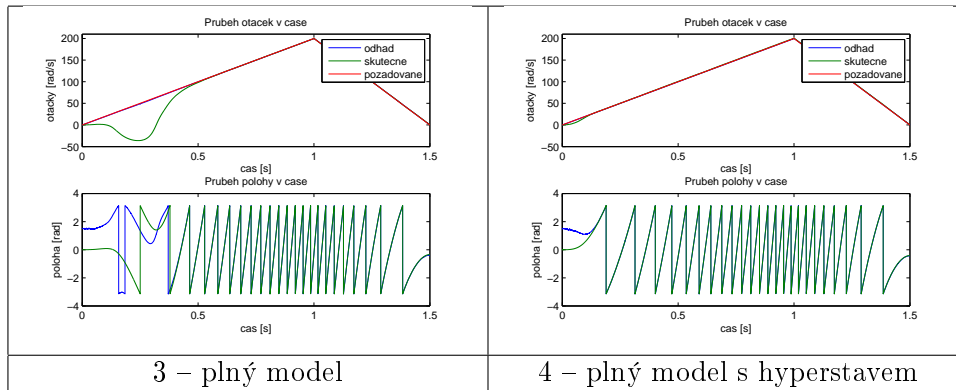
Obrázek 0.2: Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru s profilem požadovaných otáček ± 1



Obrázek 0.3: Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru s profilem požadovaných otáček ± 10



Obrázek 0.4: Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru s profilem požadovaných otáček ± 200



Obrázek 0.5: Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru s profilem požadovaných otáček ± 200 a volbou počátečního odhadu $\vartheta_0 = 1,5$ při skutečné hodnotě 0

Jednoduchý simulátor PMSM na základě rovnic

Průběhy otáček ω a polohy ϑ pro simulace na jednoduchém simulátoru zachycují grafy na obrázcích (0.1), (0.2), (0.3) a (0.4).

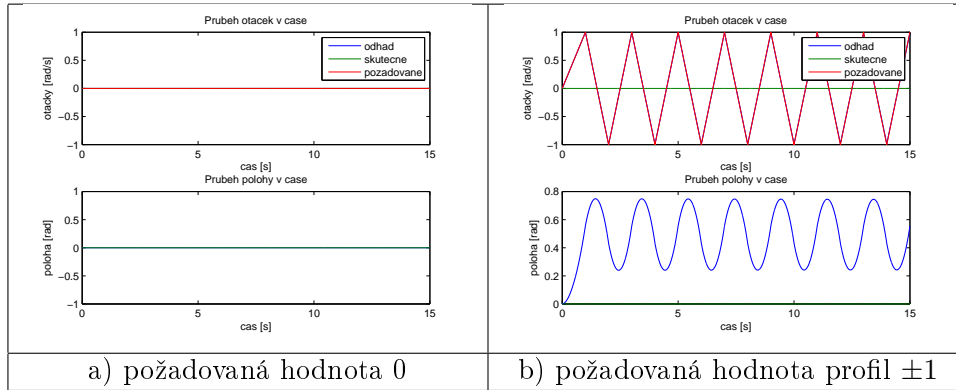
Následující tabulka shrnuje dosažené průměrné ztráty (z 10 běhů) pro jednotlivé modely a profily referenčních otáček na jednoduchém simulátoru. Jako ztráty jsou zde uvažovány pouze součty kvadrátů odchylek požadovaných a skutečných otáček.

použitý model \ požadované otáčky	0	± 1	± 10	± 200
1 – redukovaný model	2611	2731	28640	10815000
2 – redukovaný model s hyperstavem	2377	2480	3070	439740
3 – plný model	3579	3163	4268	11168
4 – plný model s hyperstavem	3240	3797	3389	61902

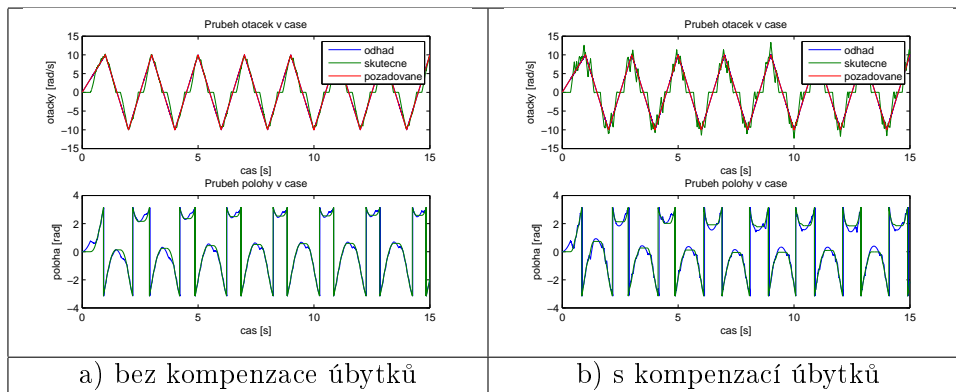
Z této tabulky je patrné, že pro redukovaný model dosahuje verze s hyperstavem nižší ztráty, což je více patrné zvláště při vyšších otáčkách. Naproti tomu nelze říci, že by verze s hyperstavem byla lepší pro plný model. Výhoda užití hyperstavu se však projeví, když máme špatný počáteční odhad polohy ϑ_0 . Uvažujme tedy počáteční odhad $\vartheta_0 = 1,5$, zatímco skutečná poloha je 0 (hodnota 1,5 je volena, protože je dostatečně daleko od 0, ale ještě nedosahuje $\frac{\pi}{2}$, kdy hrozí nebezpečí otáčení na opačnou stranu). Na grafech obrázků (0.5) je možné pozorovat počátek běhu, při profilu požadovaných otáček ± 200 , ze kterého je zřejmé lepší zvládnutí špatného počátečního odhadu polohy při užití hyperstavu. Průměrné ztráty pak jsou: $1,1035 \cdot 10^7$ pro plný model bez hyperstavu a $4,4955 \cdot 10^4$ pro plný model s hyperstavem.

Simulátor PMSM

Výsledky ze simulátoru PMSM, který více odpovídá reálnému chování stroje, již dopadly hůře. Je tomu především z důvodu zahrnutí složitějších efektů do simulátoru, jako úbytky



Obrázek 0.6: Pruběh otáček ω a polohy ϑ pro simulace na simulátoru PMSM s kompenzací úbytků napětí i bez ní pro nulové požadované otáčky a pro profil ± 1

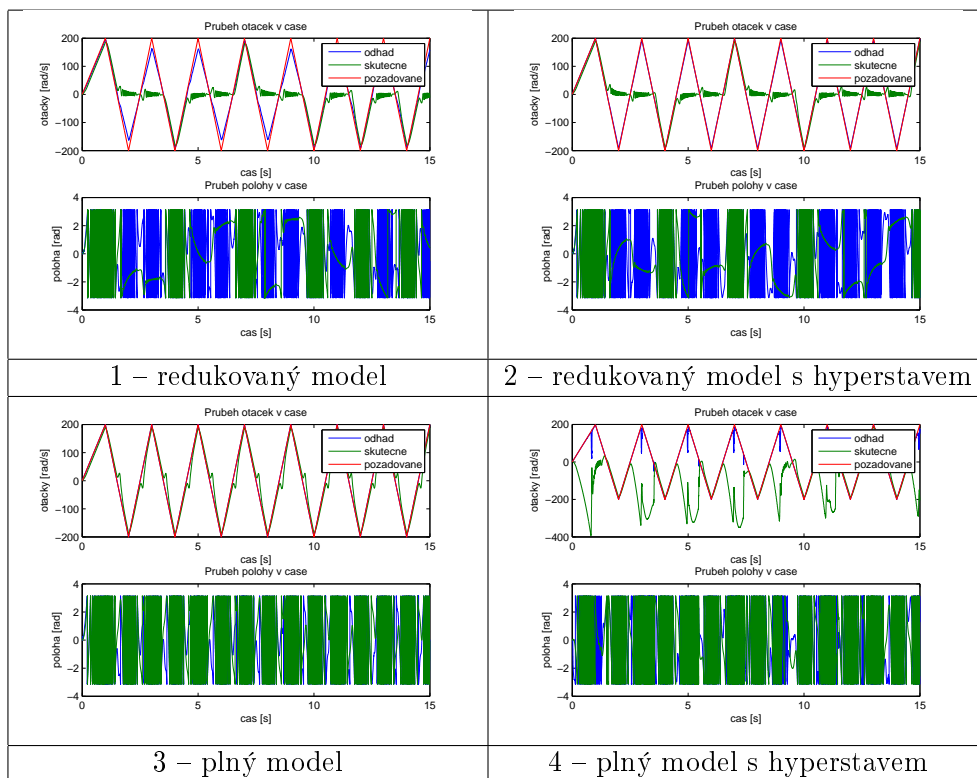


Obrázek 0.7: Pruběh otáček ω a polohy ϑ pro simulace na simulátoru PMSM s kompenzací úbytků napětí i bez ní pro reprezentativní model (3 – plný model bez hyperstavu)

napětí a mrtvé časy. Pro nulové požadované otáčky při užití kompenzace úbytků napětí i bez ní je zdnlivě vše v pořádku, viz graf na obrázku (0.6 a). Ovšem pro profil požadovaných otáček ± 1 s i bez kompenzace je výsledný průběh stejný, viz grafy na obrázku (0.6 b). Tedy nic se neděje, i když požadavek je nenulový a tento výsledek je již špatný.

Pro profil ± 10 otáčky nezůstávají nulové a dosahují požadovaných hodnot 10 respektive -10 , objevuje se zde však problém s průchody nulou, viz obrázek (0.7 a). V případě užití kompenzace úbytků napětí se situace ještě zhorší, viz obrázek (0.7 b).

V případě profilu požadovaných otáček ± 200 poskytuje simulátor PMSM lepší výsledky, je však důležité užit kompenzace úbytků napětí. Průběhy otáček ω a polohy ϑ pro simulátor PMSM bez kompenzace úbytků je zobrazen na grafch obrázek (0.8). Přínos kompenzace úbytků je pak patrný ze srovnání s grafy obrázek (0.9). Srovnání dosažených ztrát shrnuje následující tabulka:

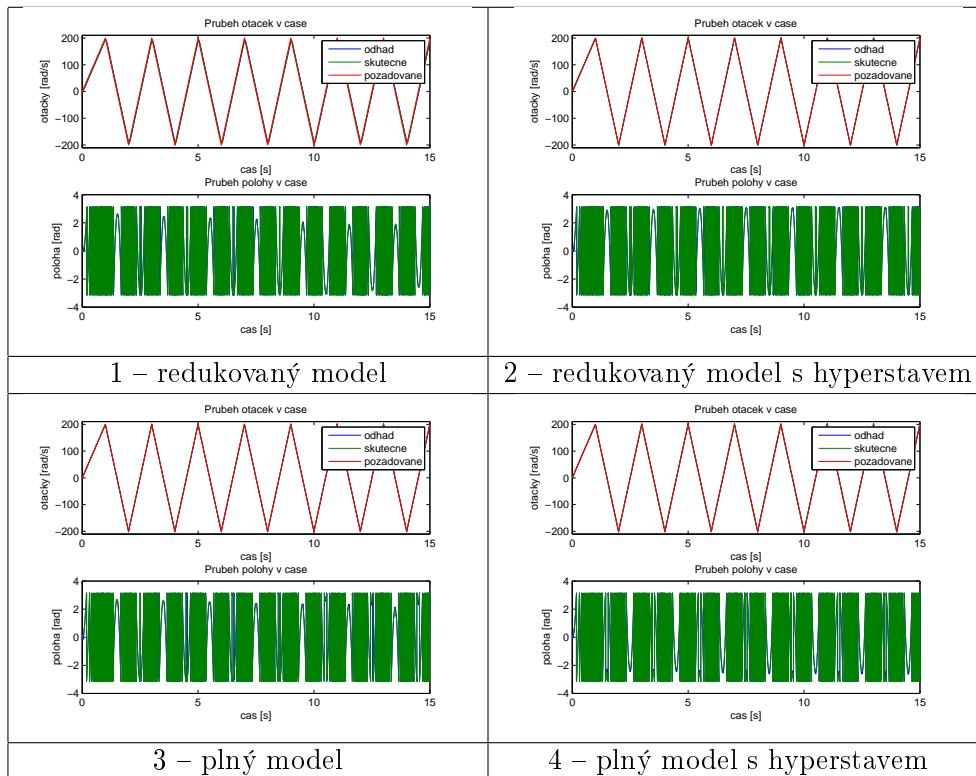


Obrázek 0.8: Průběhy otáček ω a polohy ϑ při užití simulátoru PMSM bez užití kompenzace úbytků napětí a s profilem požadovaných otáček ± 200

použitý model \ kompenzace úbytků	s kompenzací	bez kompenzace
1 – redukovaný model	$9,84 \cdot 10^8$	$8,54 \cdot 10^6$
2 – redukovaný model s hyperstavem	$1,12 \cdot 10^9$	$1,97 \cdot 10^5$
3 – plný model	$7,98 \cdot 10^7$	$2,82 \cdot 10^5$
4 – plný model s hyperstavem	$4,92 \cdot 10^9$	$9,45 \cdot 10^5$

Shrnutí

- kompenzace úbytků napětí se daří jen při vyšších otáčkách, při nižším moc nefunguje
- výhoda využití hyperstavu je především v přesnějším řízení pro redukovaný model a dále v lepším zvládnutí špatného počátečního odhadu ϑ_0
- proti špatným průchodům nulou při nízkých otáčkách zatím nic nepomáhá – ani hyperstav ani kompenzace



Obrázek 0.9: Průběhy otáček ω a polohy ϑ při užití simulátoru PMSM s užitím kompenzace úbytků napětí a s profilem požadovaných otáček ± 200