

Obsah

Obsah.....	1
1 Úvod.....	2
2 Identifikace modelu.....	3
2.1 Matematický popis.....	3
2.1.1 Motor s hlavní vrtulí.....	3
2.1.2 Motor s pomocnou vrtulí.....	3
2.1.3 Mechanická část – pohyb v elevaci.....	4
2.1.4 Mechanická část – pohyb v azimutu.....	5
2.2 Linearizace.....	5
2.3 Identifikace.....	7
3 Návrh PID regulátoru.....	11
3.1 Podle Zieglera-Nicholse.....	12
3.2 GMK.....	15
3.3 Frekvenční metody.....	18
3.3.1 PDf.....	19
3.3.2 PIDf.....	19
3.3.3 PI.....	20
4 Měření na fyzikálním modelu.....	22
5 VRML.....	24
6 VR Toolbox.....	27
6.1 Bloky VR Toolboxu.....	27
6.1.1 Joystick Input.....	27
6.1.2 VR Placeholder.....	28
6.1.3 VR Signal Expander.....	29
6.1.4 VR Sink.....	30
6.2 Propojení helikoptéry s VR Toolboxem.....	31
7 Závěr.....	32
8 Seznam použité literatury.....	33
9 Seznam použitých vývojových prostředků.....	34
10 Seznam obrázků.....	35

1 Úvod

V této bakalářské práci se budeme zabývat modelem helikoptéry od firmy HUMUSOFT viz. obr 1. Práce je rozdělena do několika částí. První část (kapitola 2) popisuje matematický model helikoptéry a její identifikaci. Ve druhé části (kapitola 3) se věnujeme návrhu PID regulátorů různými metodami a jejich simulování pomocí programu MATLAB Simulink. Ve třetí části (kapitola 4) v praxi jsou vypočítané PID regulátory ověřeny na fyzickém modelu. Čtvrtá část (kapitola 5) popisuje stručně základ jazyka VRML, bez které nelze dobře pochopit *Virtual Reality Toolbox*. Poslední pátá část (kapitola 6) je věnována popisu bloku *Virtual Reality Toolboxu* a jejich propojení se Simulinkem, demonstrace na modelu helikoptéry.

Cíl práce je vytvořit podklady pro podporu výuky předmětů věnujících se identifikací systémů, návrhu regulátorů a následné vizualizaci pomocí *Virtual Reality Toolbox*.



Obr. 1 Model helikoptéry (foto)

2 Identifikace modelu

V této kapitole se zabýváme matematickým popisem modelu a následnou identifikací.

2.1 Matematický popis

Model lze rozdělit na čtyři jednodušší podsystémy [4]

1. hlavní motor s vrtulí,
2. pomocný motor s vrtulí,
3. mechanická část – pohyb v elevaci,
4. mechanická část – pohyb v azimutu.

Zmíněné podsystémy popíšeme dále.

2.1.1 Motor s hlavní vrtulí

Vstupní proměnnou systému je napětí do hlavního motoru u_1 [V], výstupem je moment hlavní vrtule M_{hv} [N m] působící ve směru otáčení vrtule a moment M_{hvt} [N m] působící ve směru tečny vrtule. Motor lze popsat jako systém prvního řádu se vstupem u_1 a výstupem úhlovou rychlostí ω_h [rad s⁻¹]

$$\begin{aligned} \dot{x}_4 &= -\frac{1}{T_h} x_4 + \frac{1}{T_h} u_1 \\ \omega_h &= k_4 x_4 \end{aligned} \quad (1)$$

kde x_4 je stavová proměnná [s⁻¹], T_h časová konstanta vyjadřující dobu rozběhu motoru s hlavní vrtulí [s] a k_4 konstanta zesílení [-]. Vztah mezi M_{hv} a ω_h je

$$M_{hv} = \alpha_1 \omega_h |\omega_h| + \beta_1 \omega_h$$

kde α_1 a β_1 jsou konstanty. Vztah mezi M_{hvt} a ω_h je

$$M_{hvt} = \gamma_1 \omega_h |\omega_h| + \delta_1 \omega_h$$

kde γ_1 a δ_1 jsou konstanty.

2.1.2 Motor s pomocnou vrtulí

Stejně jako motor s hlavní vrtulí, i motor s pomocnou vrtulí lze popsat jako systém prvního řádu. Vstupem je zde napětí do pomocného motoru u_2 [V], výstupem je moment pomocné vrtule M_{pv} [N m] působící ve směru otáčení vrtule a M_{pvt} [N m] působící ve směru tečny vrtule. Systém lze tedy popsat vztahy

$$\begin{aligned}\dot{x}_3 &= -\frac{1}{T_p} x_3 + \frac{1}{T_p} u_2 \\ \omega_p &= k_3 x_3\end{aligned}\quad (2)$$

kde x_2 je stavová proměnná [s^{-1}], T_p časová konstanta vyjadřující dobu rozběhu motoru s pomocnou vrtulí [s] a k_2 konstanta zesílení [-], ω_h úhlová rychlost,

$$\begin{aligned}M_{pv} &= \alpha_2 \omega_p |\omega_p| + \beta_2 \omega_p \\ M_{pvt} &= \gamma_2 \omega_p |\omega_p| + \delta_2 \omega_p\end{aligned}$$

kde α_2 , β_2 , γ_2 a δ_2 jsou konstanty.

2.1.3 Mechanická část – pohyb v elevaci

Vstupní proměnné systému mechanické části – pohyb v elevaci jsou moment od hlavní vrtule M_{hv} působící ve směru otáčení vrtule, moment od pomocné vrtule M_{pvt} působící ve směru tečny vrtule a poloha těžiště daná napětím u_3 [V]. Výstupní proměnná je elevace φ [rad]. Jde tedy o rotační pohyb a pro matematický popis tohoto podsystemu můžeme vyjít z pohybové rovnice pro rotační pohyb tuhého tělesa

$$M = J\varepsilon \quad (3)$$

kde M je celkový moment síly působící na mechanickou část [N m], ε úhlové zrychlení [$rad\ s^{-2}$], a J je moment setrvačnosti [$m^2\ kg$].

Pro elevaci φ podle (3) platí

$$\ddot{\varphi} = \frac{1}{J_{el}} M \quad (4)$$

kde J_{el} je moment setrvačnosti mechanické části helikoptéry v elevaci a je funkcí polohy těžiště.

Celkový moment síly M působící na mechanickou část v elevaci je dán rovnicí

$$M = M_{hv} + M_{pvt} \cos \eta - M_g \cos \varphi - M_{trel} \quad (5)$$

kde M_g je gravitační moment mechanické části modelu při pohybu v elevaci a jeho hodnota je závislá na poloze těžiště [N m], η úhel mezi osou helikoptéry a pomyslnou přímkou procházející středem pomocné vrtule a protínající osu otáčení helikoptéry v elevaci, M_{trel} je moment způsobený třením mechanické části modelu při otáčení v elevaci a je dán vztahem

$$M_{trel} = J_{el} \delta_{el} \dot{\varphi} \quad (6)$$

kde δ_{el} je konstanta tření působící proti pohybu modelu v elevaci [s^{-1}].

Pro mechanickou část modelu pro pohyb v elevaci tedy platí dosazením rovnic (5) a (6) do rovnice (4) vztah

$$\ddot{\varphi} = \frac{M_{hv} + M_{pvt} \cos \eta - M_g(u_3) \cos \varphi - J_{el}(u_3) \delta_{el} \dot{\varphi}}{J_{el}(u_3)} \quad (7)$$

2.1.4 Mechanická část – pohyb v azimutu

Vstupní proměnné systému mechanické části – pohyb v azimutu jsou moment od pomocné vrtule M_{pv} působící ve směru otáčení vrtule, moment od hlavní vrtule M_{hvt} působící ve směru tečny vrtule, poloha těžiště daná napětím u_3 a elevace φ . Výstupní proměnná je azimut ψ [rad].

Podle rovnice (3) pro azimut ψ platí

$$\ddot{\psi} = \frac{1}{J_{az}} M \quad (8)$$

kde J_{az} je moment setrvačnosti mechanické části helikoptéry v azimutu, závislý na elevaci a na poloze těžiště. Celkový moment síly M působící na mechanickou část je dán

$$M = M_{pv} \cos(\varphi + \eta) + M_{hvt} \cos \varphi - M_{traz} \quad (9)$$

kde η je úhel mezi osou helikoptéry a pomyslnou přímkou procházející středem pomocné vrtule a protínající osu otáčení helikoptéry v elevaci, M_{traz} je moment způsobený třením mechanické části modelu při otáčení v azimutu a je dán vztahem

$$M_{traz} = J_{az} \delta_{az} \dot{\psi} \quad (10)$$

kde δ_{az} je konstanta tření působící proti pohybu modelu v azimutu [s^{-1}].

Pro mechanickou část modelu pro pohyb v azimutu tedy platí dosazením rovnic (9) a (10) do rovnice (8) vztah

$$\begin{aligned} \ddot{\psi} &= \frac{M_{pv} \cos(\varphi + \eta) + M_{hvt} \cos \varphi - \delta_{az} J_{az}(u_3) \dot{\psi}}{J_{az}(u_3)} \\ \ddot{\psi} &= \frac{(\alpha_2 (k_3 x_3)^2 + \beta_2 (k_3 x_3)) \cos(\varphi + \eta) + (\gamma_1 (k_4 x_4)^2 + \delta_1 k_4 x_4) \cos \varphi - \delta_{az} J_{az}(u_3) \dot{\psi}}{J_{az}(u_3)} \end{aligned} \quad (11)$$

2.2 Linearizace

Nyní provedeme linearizaci vztahu (11). Vstupem do systému je napětí na pomocném rotoru. Systém popíšeme stavivými rovnicemi (12).

x_1 – ψ je azimut

x_2 – úhlová rychlost helikoptéry v azimutu

x_3 – úhlová rychlost pomocného (Trial) motoru helikoptéry

x_4 – úhlová rychlost hlavního (Main) motoru helikoptéry

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \frac{(\alpha_2 (k_3 x_3)^2 + \beta_2 (k_3 x_3)) \cos(\varphi + \eta) + (\gamma_1 (k_4 x_4)^2 + \delta_1 k_4 x_4) \cos \varphi - \delta_{az} J_{az}(u_3) x_2}{J_{az}(u_3)} \\
 \dot{x}_3 &= -\frac{1}{T_p} x_3 + \frac{1}{T_p} u_2 \\
 \dot{x}_4 &= -\frac{1}{T_h} x_4 + \frac{1}{T_h} u_1
 \end{aligned} \tag{12}$$

Dynamika hlavního motoru se neuplatňuje v dynamice celého systému v pracovním bodě, proto můžeme x_4 považovat za konstantu, přestává být stavovou veličinou a tím pádem vypadá a systém se redukuje na systém 3 řádu. Jelikož φ a η jsou konstanty, rovnice se dále zjednoduší. Provedeme linearizaci:

$$\begin{aligned}
 \Delta \dot{x} &= A \Delta x + B \Delta u \\
 \Delta y &= C \Delta x + D \Delta u
 \end{aligned} \tag{13}$$

Nyní nalezneme matice A , B , C a D .

$$K_K = \frac{(2\alpha_2 k_3 x_{30} + \beta_2 k_3) \cos(\varphi + \eta)}{J_{az}(u_3)} \tag{14}$$

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\delta_{az} & K_K \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{T_p} \end{pmatrix}$$

$$C = (1 \quad 0 \quad 0)$$

$$D = (0)$$

Přenos systému se vypočítá podle rovnice (15).

$$P(s) = \{C [Is - A]^{-1} \cdot B + D\} \tag{15}$$

Po dosazení nám vyjde přenos (16).

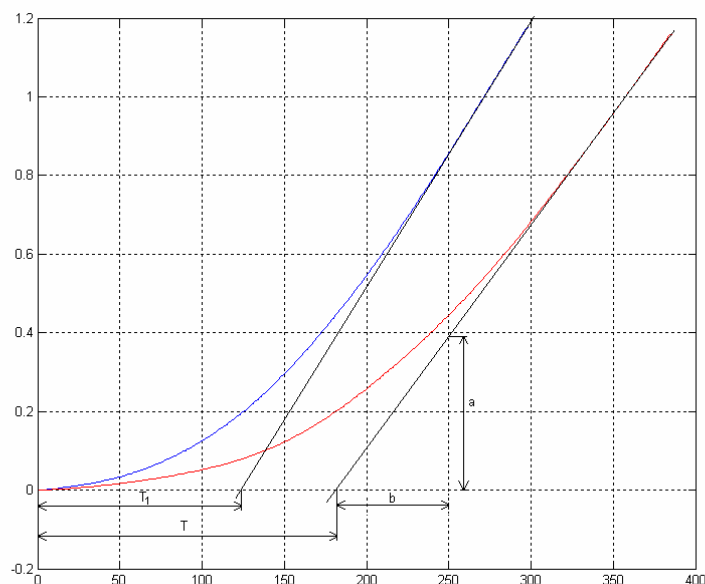
$$P(s) = \frac{\Delta Y(s)}{\Delta U(s)} = \frac{\frac{K_K}{T_p}}{s \left(s + \frac{1}{T_p} \right) (s + \delta_{az})} \tag{16}$$

2.3 Identifikace

Pracovní bod nastavíme tak, že napětí na hlavním rotoru nastavíme $U_{Main0} \in (0,1)$ a podle toho přiložíme takové napětí na zadní rotor U_{Tial0} , aby byla helikoptéra v rovnováze (U_{Tial0} musí být větší než cca 0,05V, aby se rotor vůbec roztočil, projevuje se vliv mechanického tření).

Ustálenou polohu vrtulníku jsme našli pro $U_{Main0} = 0,4V$ a $U_{Tial0} = -0,15V$.

Identifikaci jsme provedli tak, že změříme přechodovou charakteristiku systému z pracovního bobu skokem 0,07. Naměříme dvě charakteristiky, jednu „normálně“ a druhou s podržením. Helikoptéru podržíme, aby se zanedbal vliv časové konstanty zadního rotoru. Charakteristika systému s podržením je bez časové konstanty T_2 hlavního motoru. Tuto časovou konstantu následně vypočítáme z rovnice (17).



Obr. 2 Průběh helikoptéry na jednotkový skok

Modrý graf – s podržením

Červený graf – bez držení

Osa x je v počtech vzorků. Chceme-li převést na čas, musíme počet vzorků vynásobit časem potřebným na jeden vzorek, což v našem případě je $T_s = 0,01s$. Z obr. 3 odečteme následující hodnoty.

$$T = 180 \cdot 0,01 = \underline{1,8s} \quad \text{– čas bez držení}$$

$$T_1 = 125 \cdot 0,01 = \underline{1,25s} \quad \text{– čas s podržením}$$

T_2 – čas zpoždění

$$a = 0,4$$

$$b = 0,7$$

$$T = T_1 + T_2 \Rightarrow T_2 = \underline{0,45s} \quad (17)$$

K_1 – směrnice přímky (červené)

Charakteristika je naměřena pro jednotkový skok o hodnotě 0,07. Proto musíme ještě směrnici vydělit 0,07

$$K_1 = \frac{a}{0,07 \cdot b} = 8,16 \quad (18)$$

$$P(s) = \frac{K_1}{s(1+sT_1)(1+sT_2)} = \frac{K_1}{T_1T_2s\left(\frac{1}{T_1}+s\right)\left(\frac{1}{T_2}+s\right)} = \frac{\frac{K_1}{T_1T_2}}{s\left(\frac{1}{T_1}+s\right)\left(\frac{1}{T_2}+s\right)}$$

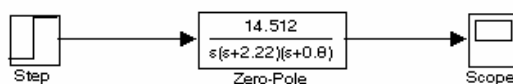
$$P(s) = \frac{K_1}{s(1+sT_1)(1+sT_2)} = \frac{8,16}{\underline{\underline{s(1+1,8s)(1+0,45s)}}} \quad (19)$$

Přenos upravíme, aby bylo možné požit v Simulinku blok Zero-Pole.

$$K = \frac{K_1}{T_1T_2} = \frac{a}{0,07 \cdot b} = \frac{0,4}{0,07 \cdot 0,7} = \underline{14,512} \quad (20)$$

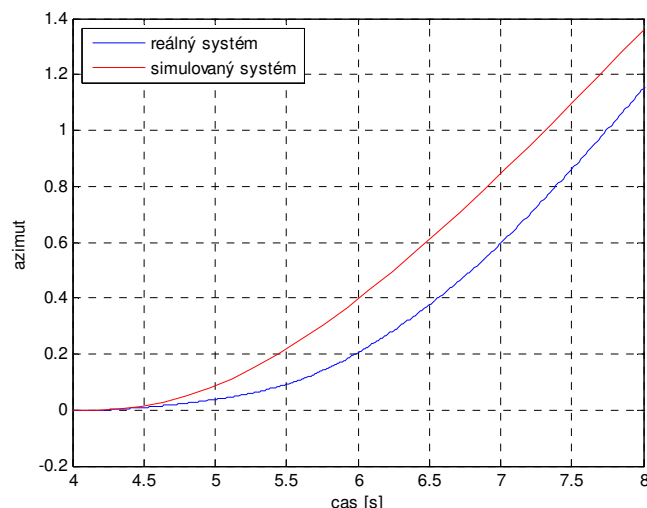
$$P(s) = \frac{K}{s\left(\frac{1}{T_1}+s\right)\left(\frac{1}{T_2}+s\right)} = \frac{14,512}{\underline{\underline{s(s+2,22)(s+0,8)}}} \quad (21)$$

Pro kontrolu použijeme Simulink, kde skok bude 0,07 a step time 4 v zapojení na obr. 3.



Obr. 3 Simulinkové schéma pro simulovaný systém

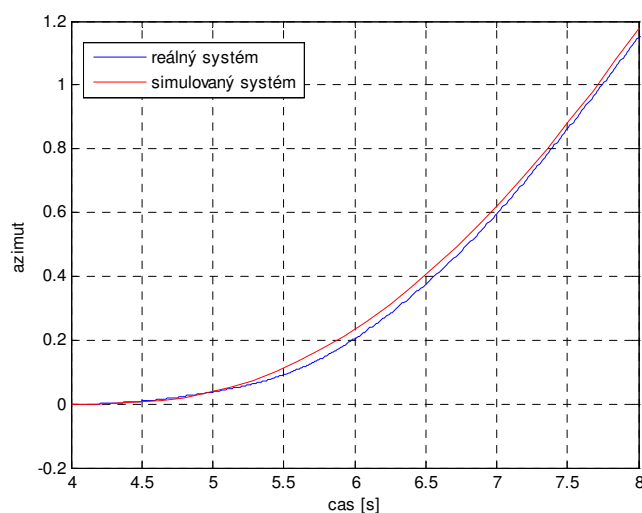
Poté výsledek zobrazíme do společného grafu s naměřenými hodnotami.



Obr. 4 Porovnání skutečného modelu a simulace

$$P(s) = \frac{14,512}{s(s+2,22)(s+0,8)} \quad (22)$$

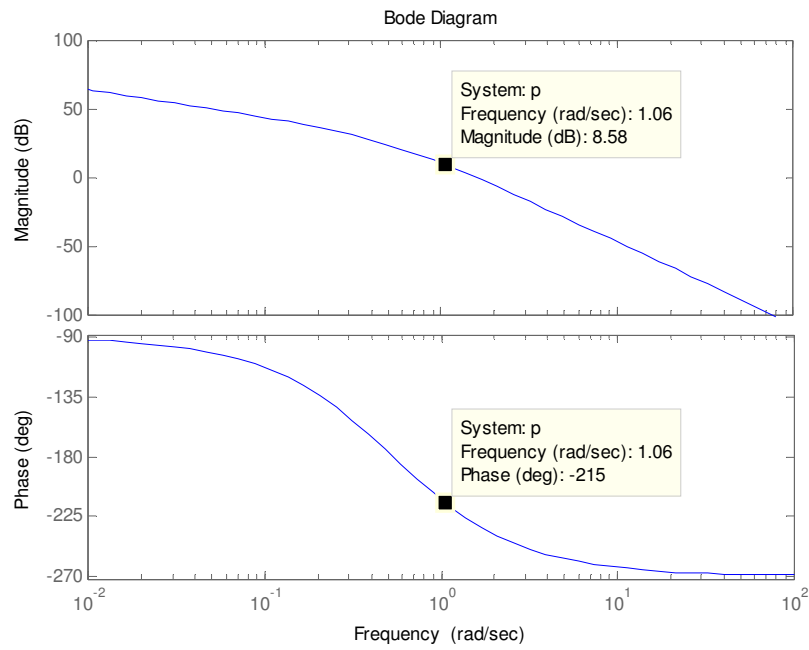
Protože průběhy si nejsou příliš podobné, musíme přenos trochu upravit. Změnou K změníme strmost neboli směrnici přímky tečny v nekonečnu a změnou T_1 nebo T_2 se změní celková časová konstanta charakteristiky.



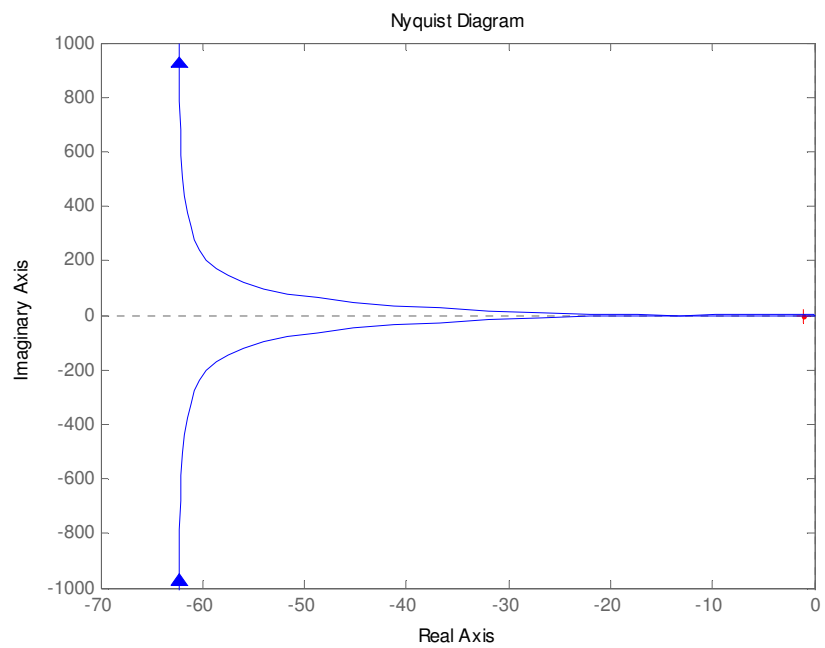
Obr. 5 Porovnání skutečného a identifikovaného modelu – upravené

Po úpravách jsme došli k přenosu

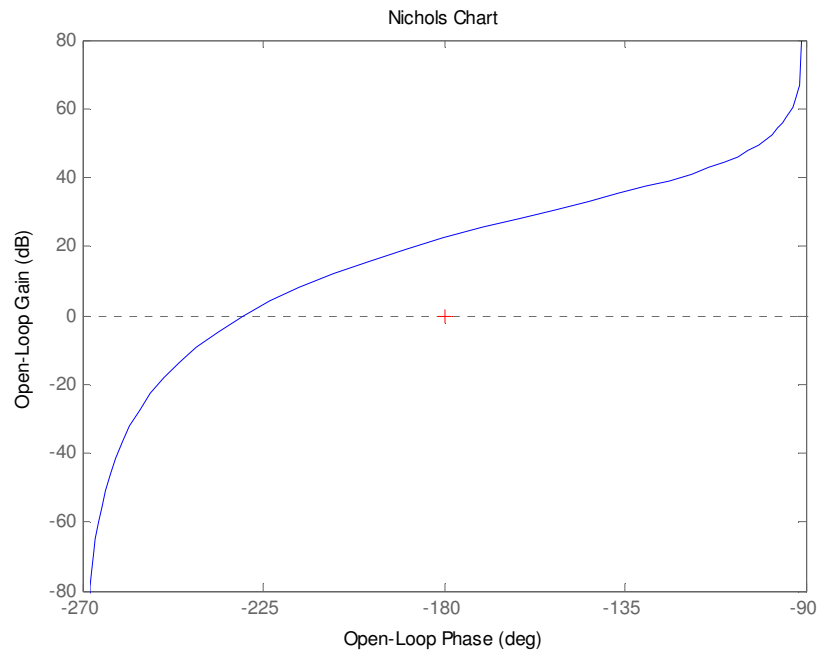
$$P(s) = \frac{4,257}{s(s+0,35)(s+0,8)} = \frac{1,19}{s(1+2,86s)(1+1,25s)} \quad (23)$$



Obr. 6 Bodeho charakteristika identifikovaného přenosu



Obr. 7 Nyquistova charakteristika identifikovaného přenosu



Obr. 8 Nicholsova charakteristika identifikovaného přenosu

3 Návrh PID regulátoru

V této kapitole budeme navrhovat PID regulátory.

Regulátor je popsán rovnicí (24)

$$C(s) = r_0 + \frac{r_{-1}}{s} + r_1 \cdot s = r_0 \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s \right) \quad (24)$$

Regulátory (obsahující D složku) se někdy doplňují filtrem (25), který zamezuje přílišnému zesilování vysokofrekvenčních šumů, takže např. PD regulátor s jednodukapacitním filtrem má přenos

$$C(s) = \frac{r_1}{T_f} \frac{s + \omega_D}{s + \omega_f} = r_0 \frac{T_D s + 1}{T_f s + 1} \quad (25)$$

kde $\omega_f = \frac{1}{T_f}$ se volí $(5-20)\omega_D$. V našem případě jsme vždy volili $\omega_f = 10\omega_D$.

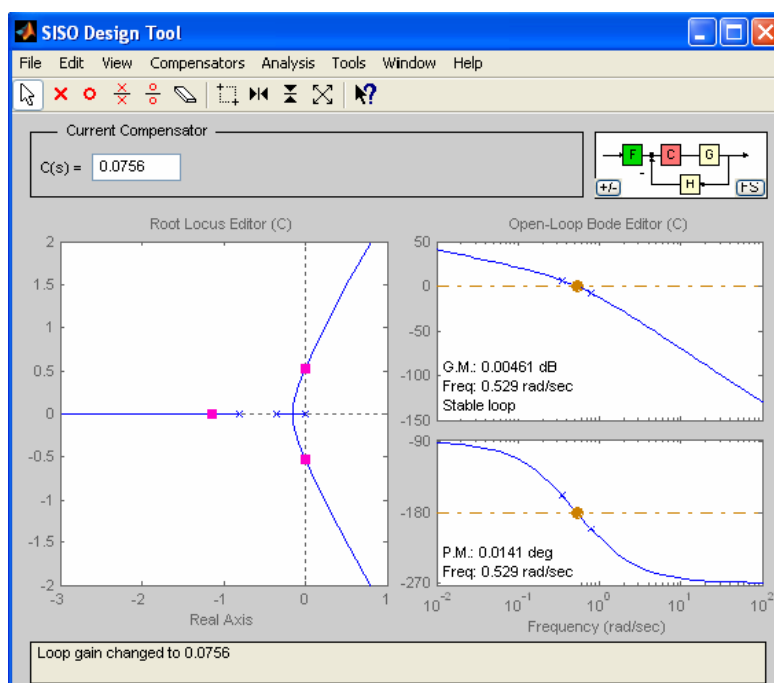
Při simulacích regulátorů v této kapitole jsme použili jednotkového skoku o velikosti 1. Výchozí hodnota regulátoru je vždy 0.

3.1 Podle Zieglera-Nicholse

Ziegler-Nicholsova metoda je založena na kritické frekvenci ω_k a kritickém zesílení r_{0k} regulované soustavy. Kritická frekvence ω_k je frekvence, při které se regulovaná soustava dostane do ustálených kmitů. Ty najdeme např. ve frekvenční charakteristice při průchodu argumentu hodnotou $\varphi = -180^\circ$ nebo experimentálně na reálné soustavě, zapojíme-li regulátor jako ryze proporcionální a zvyšujeme r_0 až do dosažení meze stability nebo výpočtem z přenosu.

Nalezení r_{0k} a T_k .

- a) pomocí SISO Design Tool v MATLABu. Přesuneme póly na mez stability a odečteme ω_k a r_{0k} .



Obr. 9 Určení kritické frekvence

$$\begin{aligned}\omega_k &= 0,529 \\ r_{0k} &= 0,0756\end{aligned}\quad (26)$$

- b) výpočtem z přenosu. Charakteristický polynom musí mít kořeny na imaginární ose:

$$\begin{aligned}s(s+0,35)(s+0,8)+4,257r_{0k} &= 0 \\ s^3+1,15s^2+0,28s+4,257r_{0k} &= 0\end{aligned}\quad (27)$$

Za s dosadíme $j\omega$.

$$\begin{aligned}(j\omega)^3+1,15(j\omega)^2+0,28j\omega+4,257r_{0k} &= 0 \\ -j\omega^3-1,15\omega^2+j0,28\omega+4,257r_{0k} &= 0\end{aligned}\quad (28)$$

Pro reálnou část a imaginární část plat:

$$-\omega^3 + 0,28\omega = 0 \tag{29}$$

$$\underline{-1,15\omega^2 + 4,257r_{0k} = 0}$$

$$\omega(-\omega^2 + 0,28) = 0 \Rightarrow \underline{\underline{\omega_k = 0,529}} \tag{30}$$

$$\omega_1 = 0$$

$$\underline{\underline{r_{0k} = 0,0756}}$$

$\omega = 0$ v tomto případě lze z fyzikálního hlediska vyloučit.

$$\omega_k = \frac{2\pi}{T_k} \Rightarrow T_k = \underline{\underline{11,88s}} \tag{31}$$

Nastavíme konstanty regulátoru podle jeho typu viz. tabulka 1.

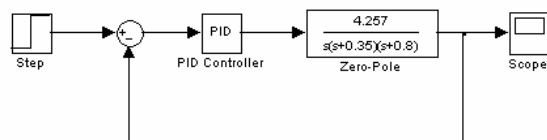
Regulátor	P	PI	PID	PD
r_0	$0,5r_{0k}$	$0,45r_{0k}$	$0,6r_{0k}$	$0,45r_{0k}$
T_I	-	$0,85T_k$	$0,5T_k$	-
T_D	-	-	$0,12T_k$	$0,12T_k$

Tabulka 1 Nastavení regulátoru podle kritického zesílení a kritické frekvence

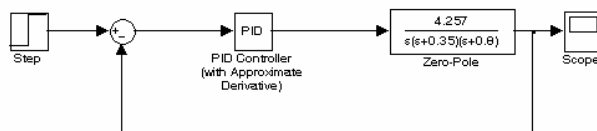
Po dosazení vyjde tabulka 2.

Regulátor	P	PI	PID	PD
r_0	0,0378	0,034	0,045	0,034
T_i	-	10,1	5,94	-
T_d	-	-	1,426	1,426

Tabulka 2 Nastavení regulátoru po dosazení hodnot



Obr. 10 Simulinkové schéma pro měření regulátoru bez filtru



Obr. 11 Simulinkové schéma pro měření regulátoru s filtrem

Přenos pro **P** regulátor

$$C_P(s) = r_0 = 0,0378 \quad (32)$$

Přenos pro **PI** regulátor

$$C_{PI}(s) = r_0 + \frac{r_0}{T_i s} = 0,034 + \frac{0,0034}{s} \quad (33)$$

Přenos pro **PD** regulátor

$$C_{PD}(s) = r_0 + r_0 T_d s = 0,034 + 0,048s \quad (34)$$

Přenos pro **PDF** regulátor

$$T_f = 0,1 \cdot T_D$$

$$C_{PDF}(s) = \frac{r_0 + r_0 T_d s}{T_f s + 1} = \frac{0,048s + 0,034}{0,1426s + 1} \quad (35)$$

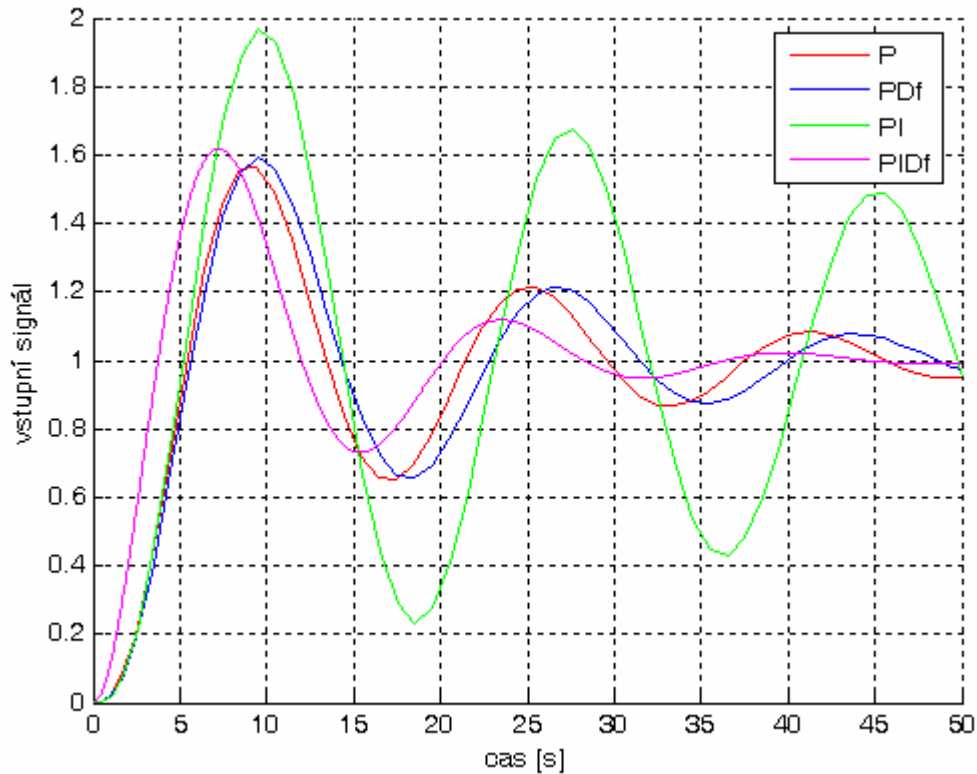
Přenos pro **PID** regulátor

$$C_{PID}(s) = r_0 + \frac{r_0}{T_i s} + r_0 T_d s = 0,045 + \frac{0,0076}{s} + 0,0642s \quad (36)$$

Přenos pro **PIDf** regulátor

$$T_f = 0,1 \cdot T_D$$

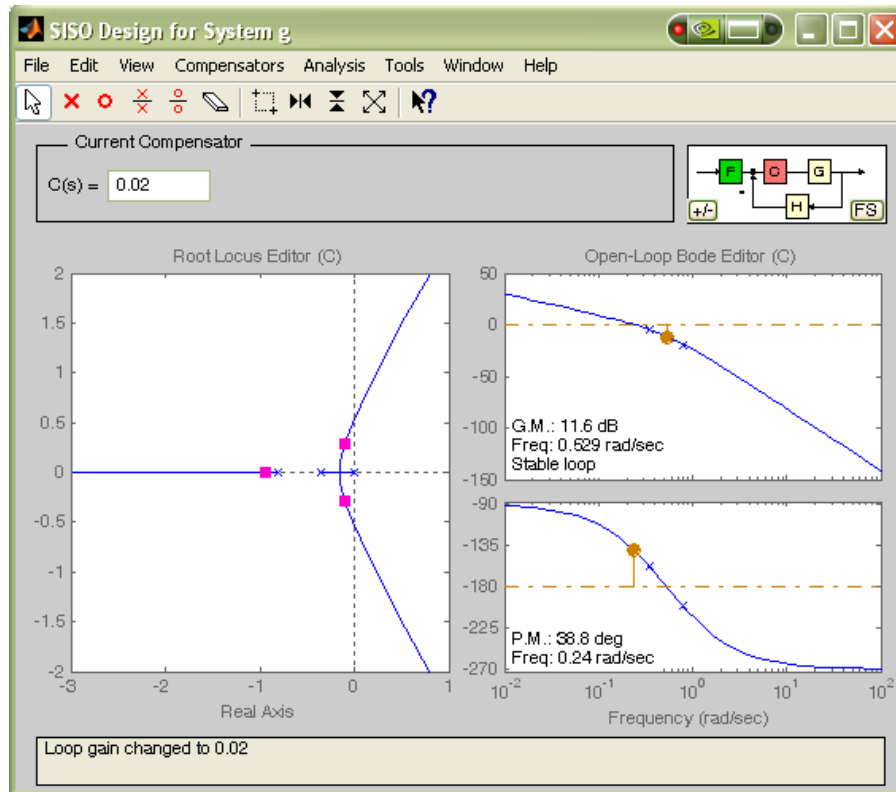
$$C_{PIDf}(s) = \frac{r_0 + \frac{r_0}{T_i s} + r_0 T_d s}{T_f s + 1} = \frac{0,045 + \frac{0,0076}{s} + 0,0642s}{0,1426s + 1} = \frac{0,0642s^2 + 0,045s + 0,0076}{0,1426s^2 + s} \quad (37)$$



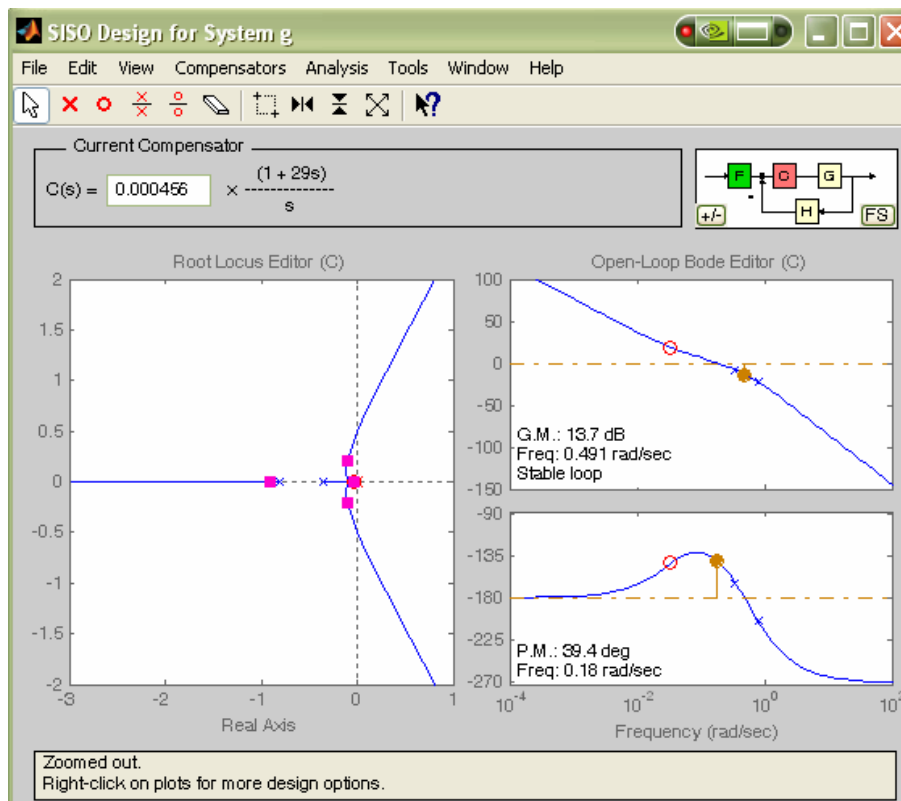
Obr. 12 Porovnání regulátorů navržených Ziegler-Nicholsovou metodou

3.2 GMK

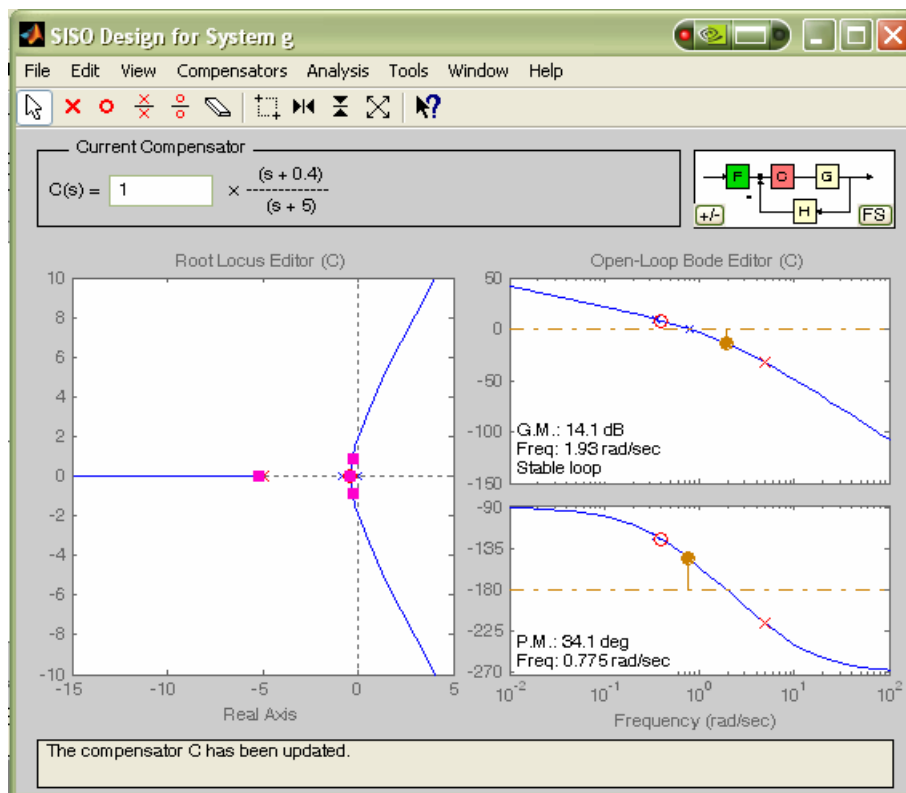
Metoda geometrického místa kořenů (GMK, root locus) vychází z pólů a nul otevřeného regulačního obvodu a na jejich základě určuje polohu pólů uzavřeného regulačního obvodu v závislosti na změně zesílení otevřeného regulačního obvodu. Pomocí jednoduchých pravidel a trochy zkušenosti volíme polohu pólů a nul regulátoru.



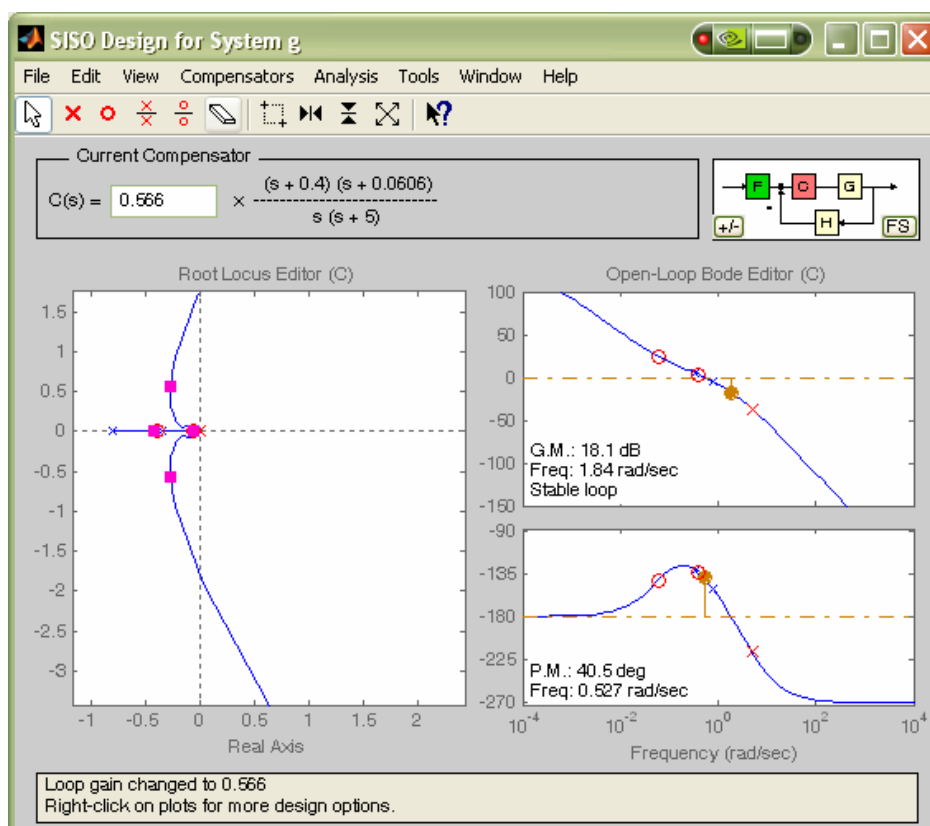
Obr. 13 Geometrické místo kořenů pro P regulátor



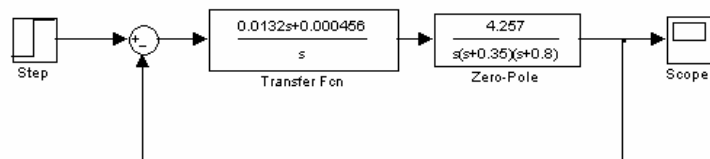
Obr. 14 Geometrické místo kořenů pro PI regulátor



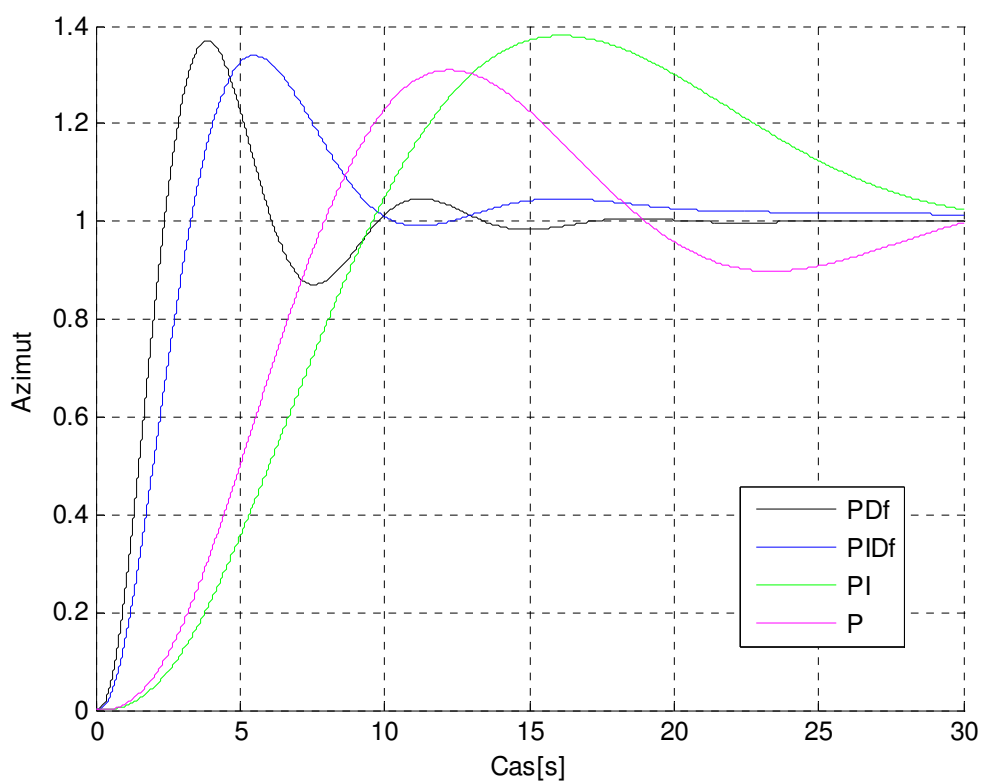
Obr. 15 Geometrické místo kořenů pro PDf regulátor



Obr. 16 Geometrické místo kořenů pro PIDf regulátor



Obr. 17 Schéma použité v simulinku pro simulaci



Obr. 18 Porovnání regulátorů navržených GMK metodou

3.3 Frekvenční metody

Frekvenční metody návrhu se snaží upravit frekvenční charakteristiku otevřené smyčky $G(j\omega)$ tak, aby výsledná frekvenční charakteristika uzavřené smyčky $F(j\omega)$ měla požadovaný průběh. Mezi $G(j\omega)$ a $F(j\omega)$ existuje totiž jednoznačný vztah:

$$F(j\omega) = \frac{G(j\omega)}{1 + G(j\omega)} \quad (38)$$

Fázová bezpečnost (phase margin, PM) říká, o jak velké zpoždění (fázi) si můžeme dovolit zpozdít vstupní signál na frekvenci ω_{PM} , pro kterou má přenos $G(s)$ jednotkové zesílení, než se fáze obrátí na -180° a zpětnovazební systém se tak dostane na hranici stability.

V našem případě volíme fázovou bezpečnost tak, aby nám vyšla proporcionální složka regulátoru dostatečně velká. Fyzicky model dokáže reagovat na proporcionální složku o min. hodnotě 0,1, přičemž dokáže rozlišit pouze rozdíly o velikosti 0,05.

3.3.1 PDF

Zvolíme $PM = 45^\circ$ a z grafu (viz obr. 7) odečteme ω_D a Au.

$$PM = 10^\circ$$

$$-180^\circ + PM - 45^\circ = -180^\circ + 45^\circ - 45^\circ = -215^\circ$$

$$\omega_D = \underline{1,06}$$

$$Au = 8,58dB \Rightarrow 10^{\frac{8,58}{20}} = \underline{2,69}$$

$$k_D = \frac{1}{\sqrt{2} \cdot \omega_D |P(j\omega)|} = \frac{1}{\sqrt{2} \cdot 1,06 \cdot 2,69} = \underline{0,248}$$

$$k_P = \frac{1}{\sqrt{2} |P(j\omega)|} = \frac{1}{\sqrt{2} \cdot 2,69} = \underline{0,263}$$

$C_{PD}(s)$ - regulátor bez filtru

$C_{PDF}(s)$ - regulátor s filtrem

$$C_{PD}(s) = k_D s + k_P = k_P (T_D s + 1) = 0,248s + 0,263$$

$$\omega_f = \frac{1}{T_f} = 10\omega_D \Rightarrow T_f = \frac{1}{10\omega_D} = \underline{0,09}$$

$$C_{PDF}(s) = \frac{k_D s + k_P}{T_f s + 1} = \frac{0,248s + 0,263}{\underline{\underline{0,09s + 1}}} \quad (39)$$

3.3.2 PIDf

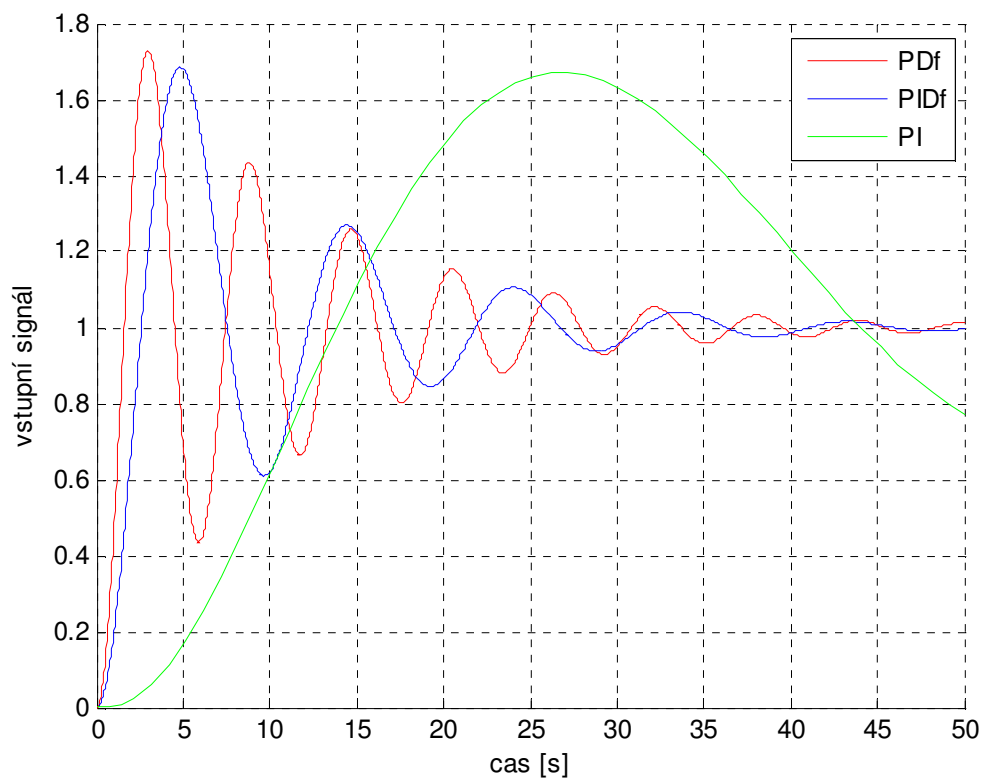
Zvolíme $PM = 39,3^\circ$ a z Bodeho grafu (obr. 7) odečteme ω_D a Au.

$$\begin{aligned}
 PM &= 39,3^\circ \\
 &-180^\circ \\
 \omega_D &= \underline{0,531} \\
 Au &= 22,3dB \Rightarrow 10^{\frac{22,3}{20}} = \underline{13,03} \\
 k_D &= \frac{1}{\sqrt{2} \cdot \omega_D |P(j\omega)|} = \frac{1}{\sqrt{2} \cdot 0,531 \cdot 13,03} = \underline{0,102} \\
 k_P &= 1,1\omega_D k_D = 1,1 \cdot 0,531 \cdot 0,102 = \underline{0,087} \\
 k_I &= 0,1\omega_D^2 k_D = 0,1 \cdot 0,531^2 \cdot 0,102 = \underline{0,0096} \\
 C_{PID}(s) &= k_D s + k_P + \frac{k_I}{s} = \frac{k_D s^2 + k_P s + k_I}{s} \\
 C_{PID}(s) &= \frac{0,102s^2 + 0,087s + 0,0096}{s} \\
 \omega_f &= \frac{1}{T_f} = 10\omega_D \Rightarrow T_f = \frac{1}{10\omega_D} = 0,188s \\
 C_{PIDf}(s) &= \frac{k_D s^2 + k_P s + k_I}{s(T_f s + 1)} = \frac{0,102s^2 + 0,087s + 0,0096}{s(0,188s + 1)} \quad (40)
 \end{aligned}$$

3.3.3 PI

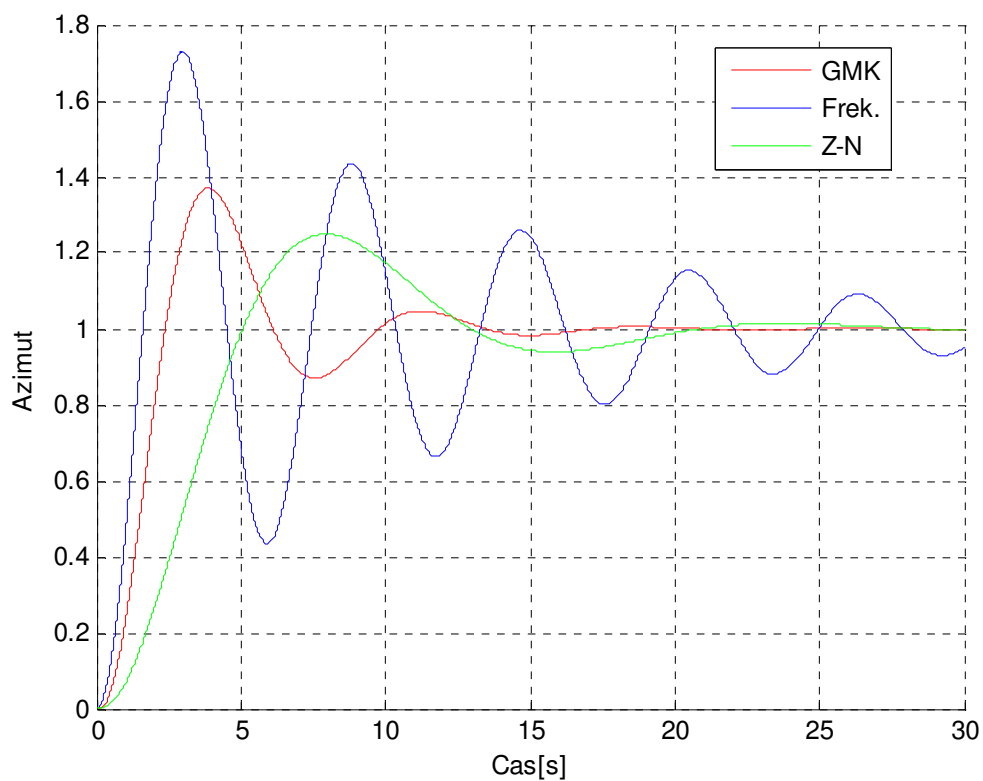
Zvolíme $PM = 20^\circ$ a z Bodeho grafu (obr. 7) odečteme ω_D a Au .

$$\begin{aligned}
 PM &= 20^\circ \\
 -90^\circ + PM - 45^\circ &= -90^\circ + 20^\circ - 45^\circ = -175^\circ \\
 \omega_I &= \underline{0,485} \\
 Au &= 42,5dB \Rightarrow 10^{\frac{42,5}{20}} = \underline{131,8} \\
 k_I &= \frac{\omega_I}{\sqrt{2} |P(j\omega)|} = 0,00058 \\
 k_P &= \frac{1}{\sqrt{2} |P(j\omega)|} = 0,0054 \\
 C_{PI}(s) &= k_P + \frac{k_I}{s} = 0,0054 + \frac{0,00058}{s} \quad (41)
 \end{aligned}$$



Obr. 19 Porovnání regulátorů navržených frekvenční metodou

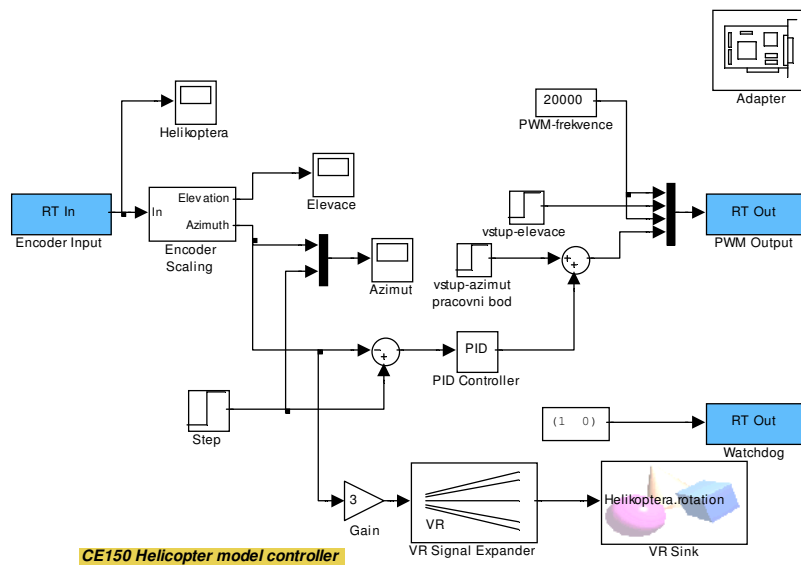
Pro zobrazení grafu jsme použili stejné bloky v Simulinku jako u Z-N metody.



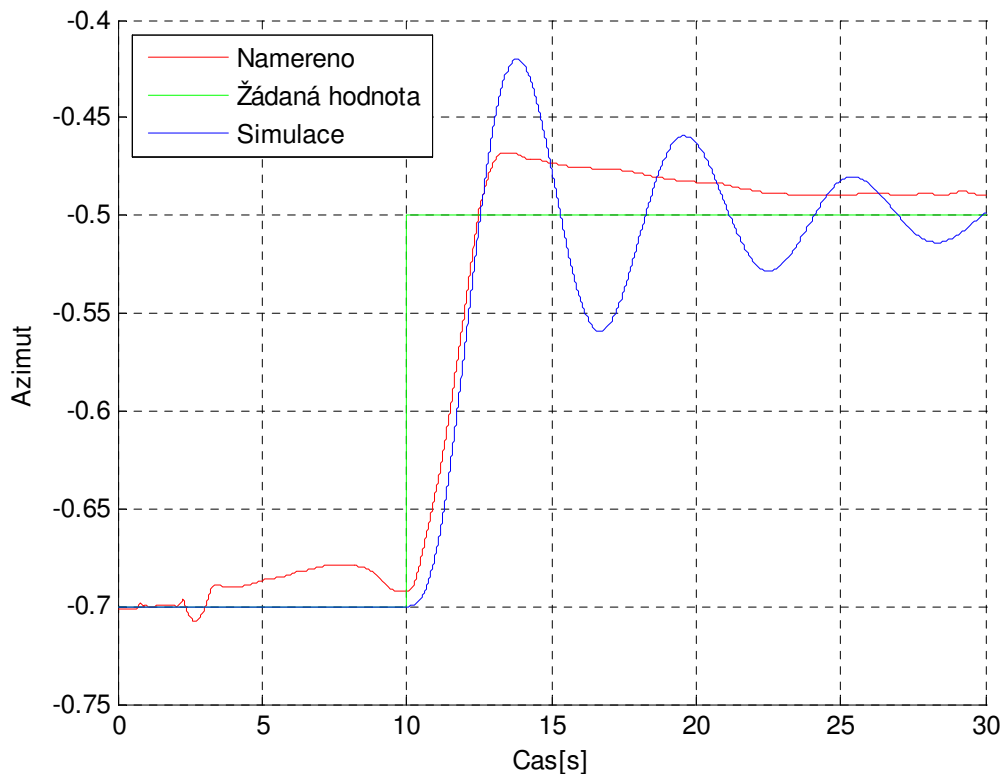
Obr. 20 Porovnání PDf regulátorů

4 Měření na fyzikálním modelu

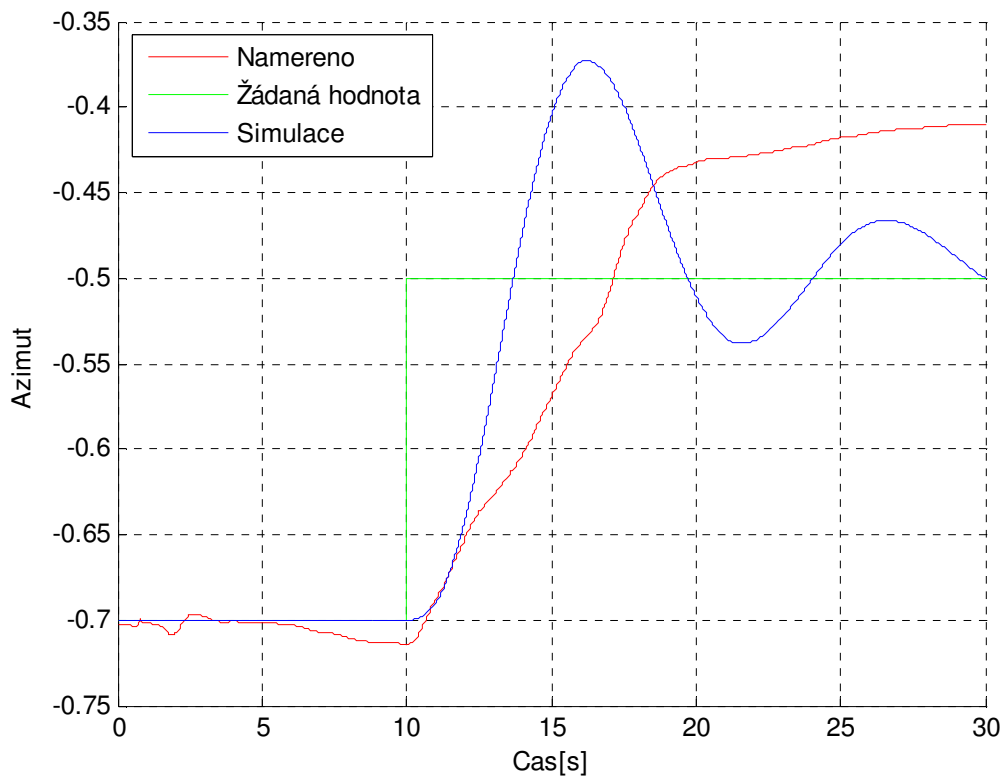
Nyní vypočítané regulátory ověříme prakticky. Zapojení v Simulinku je na obr. 22. Jelikož model dokáže rozlišit k_p a k_D o minimální velikosti cca 0,1 použijeme pouze ty navržené regulátory, které to splňují. Jsou to jsou PD (obr. 23) a PID (obr.24) regulátory navržené frekvenční metodou. Na obr. 25 je použita metoda cyklická variace konstant regulátoru. Hodnoty regulátoru (obr. 25) jsou PD jsou $k_p = 0,8$ a $k_D = 0,6$. Hodnoty regulátoru PID jsou $k_p = 0,8$ $k_D = 0,6$ a $k_I = 0,2$.



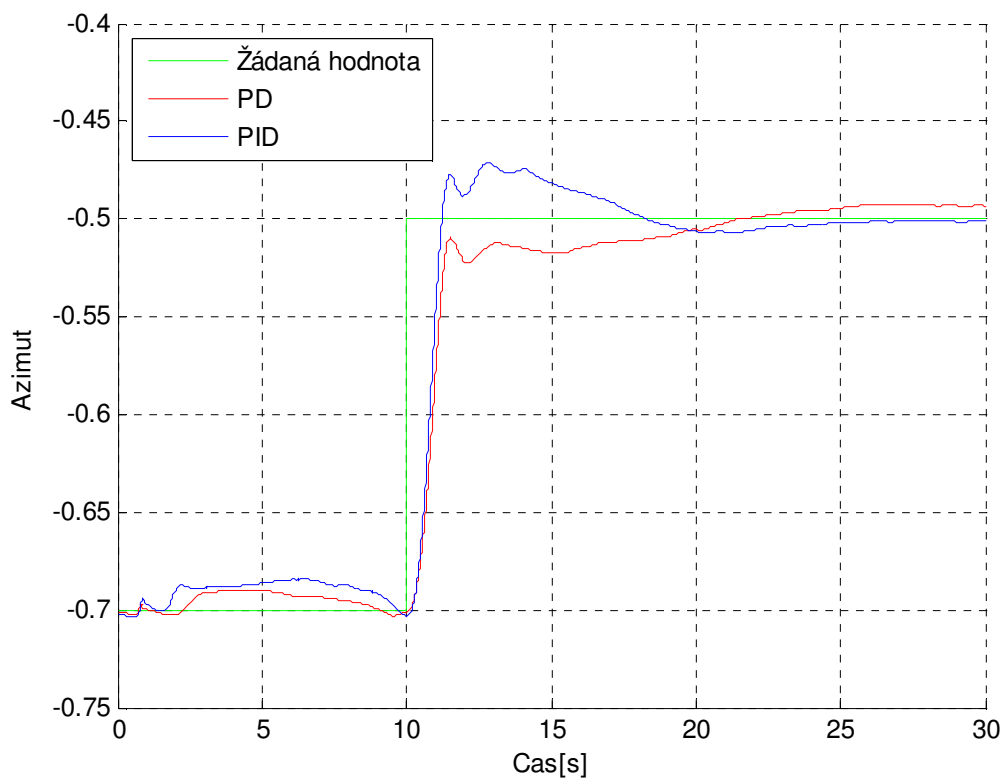
Obr. 21 Simulinkové schéma pro měření modelu



Obr. 22 PD regulátor navržený frekvenční metodou



Obr. 23 PID regulátor navržený frekvenční metodou



Obr. 24 Regulátory navržené metodou cyklické variace konstant

5 VRML

VRML (*Virtual Reality Modelling Language*) je určen pro popis trojrozměrných scén v prostředí WWW. Lze je prohlížet pomocí webových prohlížečů s příslušným pluginem (např. Internet Explorer, Mozilla FireFox). 3D scény se ukládají ve formě textových souborů jako program napsaný v programovacím jazyce VRML (podobně jako např. HTML). S pomocí VRML lze popsat libovolné těleso v prostoru, včetně vlastností povrchu tělesa i okolí (světla, zvuky). VRML 97 je de facto standardem pro prezentaci trojrozměrných dat na internetu, specifikace jazyka byla přijata jako mezinárodní norma ISO.

Popis scény je jakýmsi stromem s hierarchickou strukturou a samotné definice objektů jsou v uzlech a listech tohoto stromu (všem se říká uzly). Některé uzly tvoří rodiče pro jiné uzly, jiné uzly zase mohou být pouze potomky těchto uzlů.

Uzly VRML je možné podle své funkce rozdělit do dvou hlavních skupin, na *statické* (popisují tvar, vzhled, světla, textury, zvuky) a *dynamické*.

Statické uzly lze dále dělit na:

- **geometrické**
popis geometrických parametrů - tvaru a velikosti objektů
- **vlastností**
definice vlastností objektů - normály, souřadnice
- **vzhledu**
popis povrchu objektů - materiály, textury
- **světlo a zvuk**
definice různých druhů světla a zvuků
- **speciální**
popisují uzly pro zvláštní použití - odkaz, úroveň detailu atd.
- **skupinové**
uzly sdružující jiné uzly do skupin

Dynamické uzly se vyznačují tím, že jsou schopny generovat takové události, které lze buď přímo nebo zprostředkovaně použít ke změně stavu ostatních uzlů.

Zde je příklad scény zapsané ve VRML - představuje jednoduchý model vozíku:

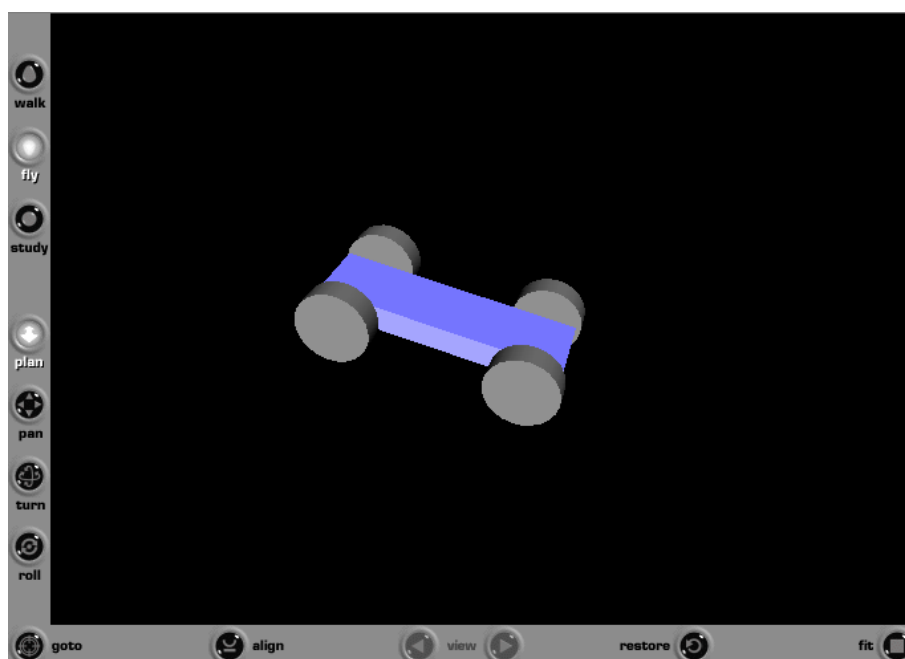
```
#VRML V2.0 utf8
Shape
{
    geometry Box { size 1 .1 .3 }
    appearance Appearance
    {
        material Material { emissiveColor 0 0 1 }
    }
}
Transform
{
    rotation 1 0 0 1.570796
    translation .375 0 .25
    children DEF KOLO Shape
    {
        geometry Cylinder
        {
            height .1
            radius .15
        }
        appearance Appearance
    }
}
```



```

        {
            material Material
            {
                diffuseColor .7 .7 .7
            }
        }
    }
Transform
{
    rotation 1 0 0 1.570796
    translation .375 0 -.25
    children USE KOLO
}
Transform
{
    rotation 1 0 0 1.570796
    translation -.375 0 -.25
    children USE KOLO
}
Transform
{
    rotation 1 0 0 1.570796
    translation -.375 0 .25
    children USE KOLO
}

```



Obr. 25 Scéna zobrazená prohlížečem Cortona

VRML soubor začíná hlavičkou (první řádek) a pokračuje popisem scény. Každá scéna se skládá z tzv. uzlů (jejich identifikátory začínají velkým písmenem); v našem příkladu se vyskytují dva nejpoužívanější typy uzlů, a sice *Shape* a *Transform*. První z nich reprezentuje geometrické útvary (kvádry, válce, koule, kužele nebo uživatelem definovaná tělesa), druhý popisuje transformace těles - posunutí, otočení a změnu měřítka. Vlastnosti uzlů určují jejich parametry (názvy parametrů začínají malým písmenem) - mohou to být různé číselné údaje, vektory, textové řetězce nebo další uzly. Například uzel typu *Shape* má parametry *geometry* (udává typ a rozměry tělesa) a *appearance* (popisuje vzhled tělesa). Zápis

```
Shape
{
    geometry Cylinder
    {
        height .1
        radius .15
    }
    appearance Appearance
    {
        material Material
        {
            diffuseColor .7 .7 .7
        }
    }
}
```

představuje šedý (barvy se zapisují v RGB) válec s poloměrem podstavy 0,15 m a výškou 0,1 m. Osa válce splývá s osou z soustavy souřadnic, těžiště válce je v jejím počátku. Chceme-li válec použít jako kolo vozíku, musíme jej otočit o 90 stupňů (1.57 radiánu) kolem osy x (vektor 1 0 0):

```
Transform
{
    rotation 1 0 0 1.570796
    children Shape
    {
        geometry Cylinder
        {
            height .1
            radius .15
        }
        appearance Appearance
        {
            material Material
            {
                diffuseColor .7 .7 .7
            }
        }
    }
}
```

(Kola je dále potřeba ještě posunout na správné místo - viz parametr *translation* v první ukázce.)

Transformace se aplikuje na všechny uzly obsažené v parametru *children* - v našem případě na uzel *Shape*, obecně jich smí být více. Může se zde objevit i další uzel *Transform*; takto lze vytvořit celou hierarchii transformací, které se skládají. Uzly z parametru *children* se označují jako potomci daného rodičovského uzlu. VRML scény se někdy kreslí jako stromy znázorňující jejich hierarchii.

Dosud jsme nevysvětlili význam klíčových slov `DEF` a `USE`, která se rovněž objevila v naší ukázce. Konstrukce `DEF JMENO Uzel { ... }` slouží k pojmenování uzlu. Na takové uzly se potom můžeme odkazovat na jiných místech VRML souboru. V našem příkladu jsme použili konstrukci `USE KOLO`, která vytvoří kopii uzlu `KOLO`. Všechna kola vozíku jsou totiž stejná, a tak stačí popsat jen jedno z nich (ostatní tři jsou kopie). Tím se zkrátí délka souboru, a navíc úprava jednoho kola způsobí stejnou změnu u všech ostatních.

Jazyk VRML zahrnuje kromě *Shape* a *Transform* řadu dalších užitečných uzlů.

6 VR Toolbox

VR Toolbox (*Virtual Reality Toolbox*) umožňuje obousměrnou interakci prostředí MATLAB/Simulink s prostředím virtuální reality. Modely dynamických systémů vytvořené v Simulinku je možné realisticky vizualizovat v trojrozměrném prostředí a získat tak lepší prostorovou představu o tom, jak fungují. Během simulace se uživatel může pohybovat ve virtuální scéně, pozorovat simulovaný systém z různých pohledů, být jeho součástí a dokonce jej z prostředí virtuální reality ovládat. Při konstrukci zařízení lze na virtuálním modelu ověřit, jakým způsobem interagují jejich součásti (Virtual Prototyping). Virtual Reality Toolbox má i objektově orientované programovací rozhraní, pomocí kterého je možné pracovat s virtuálními světy v prostředí MATLABu. Tento způsob práce je vhodný pro vizualizaci složitých geometrických tvarů, výsledků simulace fyzikálních dějů (MKP výpočty), morphing a podobně.

Ve spojení s produktem Real Time Windows Target umožňuje Virtual Reality Toolbox 3D interakci s modely pracujícími v reálném čase. Virtual Reality Toolbox je jedním ze standardních grafických výstupů pro vizualizaci mechanických soustav. Virtual Reality Toolbox obsahuje rovněž bloky pro připojení speciálních periférií pro interakci modelů v Simulinku s prostředím virtuální reality - Joystick, SpaceMouse a pod. Virtual Reality Toolbox je založen na otevřeném standardu VRML 97 (Virtual Reality Modelling Language), což umožňuje snadnou tvorbu virtuálních světů a publikování výsledků v síti WWW. Produkt lze využít ke studiu chování fyzikálních a biologických systémů v celé řadě oborů.

6.1 Bloky VR Toolboxu

Zařízení vstupního řízení

Joystick Input zpracuje vstupní signál od zařízení asynchronní ovládací páky

VRML související signály

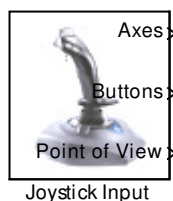
VR Placeholder pošle nespecifikovanou hodnotu do bloku Virtual Reality Toolbox

VR Signal Expander rozšiřuje vstupní vektory do úplně schopného VRML pole vektorů

Virtuální světy

VR Sink zapíše data ze Simulinkového modelu do virtuálního světa

6.1.1 Joystick Input



Účel Zpracování z asynchronního zařízení ovládací páky

Popis Blok Joystick Input poskytuje vhodnou interakci mezi Simulinkovým modelem virtuálním světem přidruženým s blokem Virtual Reality Roolbox. Funguje pouze v operačních systémech Windows-

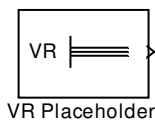
Blok Joystick Input používá osy, tlačítka a selektor point-of-view, pokud je přítomen. Tento blok může být používán jako kterýkoliv jiný simulinkový blok typu source (zdroj). Jeho výstupní porty ukazují stav ovládní joysticku v osách a tlačítkách.

Blok Joystick Input podporuje i zařízení se zpětnou vazbou (force-feedback).

Joystick ID – Identifikace přiřazená ovládací páce systémem. Tyto vlastnosti můžete najít v sekci herní zařízení systému v Ovládacích panelech.

Upřesnění I/O portů podle schopnosti ovládací páky – Jestli tuto volbu škrtnete, porty bloku nemají plnou šířku poskytnutou interfací Herní zařízení. Namísto Virtual Reality Toolbox dynamicky upřesní porty tak, aby odpovídaly schopnostem připojené ovládací páky.

6.1.2 VR Placeholder

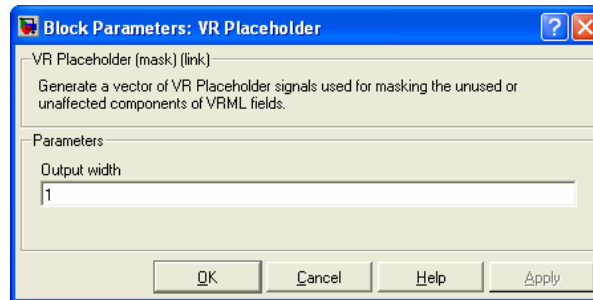


Účel Posílá nespecifikované hodnoty do bloků VR Toolboxu

Popis Blok VR Placeholder odesílá speciální hodnotu, která je interpretována blokem VR Sing jako nespecifikovaná. Když se tato hodnota objeví na vstupu bloku VR Sing (je jedno, zda jako jednoduchá hodnota nebo jako část vektoru) příslušná hodnota ve virtuálním světě zůstává nezměněna. Použijeme tento blok ke změně pouze jedné hodnoty z velkého vektoru, např. se tento blok dá použít ke změně pouze jedné souřadnice ve 3-D pozici.

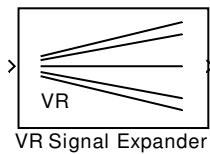
Výstupní hodnota bloku VR Placeholder by neměla být modifikována dříve, než bude použita v ostatních blocích VR

Výstup Výstupní signály bloku VR Placeholder jsou typu double



Obr. 26 Dialogové okno Placeholder

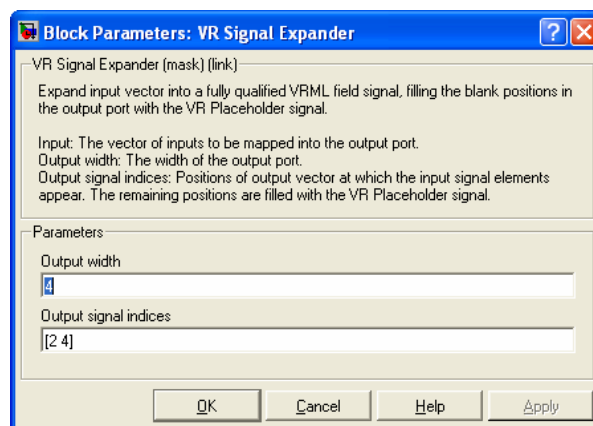
6.1.3 VR Signal Expander



Účel Rozšíří vstupní vektor na přesně stanovené VRML pole vektorů

Popis VR Signal Expander vytvoří vektor o předefinované délce (užitím některých hodnot z vstupních veličin) a splní reset při placeholder signálů.

Výstup VR Signal Expander blok akceptuje a výstupní signál je typu double



Obr. 27 Dialogové okno Expander

Popis obr. 27:

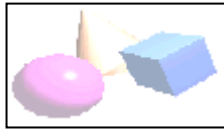
Output width – Jak dlouhý bude výstupní vektor

Output signal indices – Vektor indikuje pozici, který vstupní signál se objeví na výstupu

Například: předpokládejme, že chceme vstupní vektor s dvěma signály a výstupní se čtyřmi signály a přejeme si první signál na pozici 2 a druhý signál na pozici 4.

V editovacím poli *Output width* je 4 a v editovacím poli *Output signal indices* je [2,4]. První a třetí výstupní signál je nspecifikovaný.

6.1.4 VR Sink

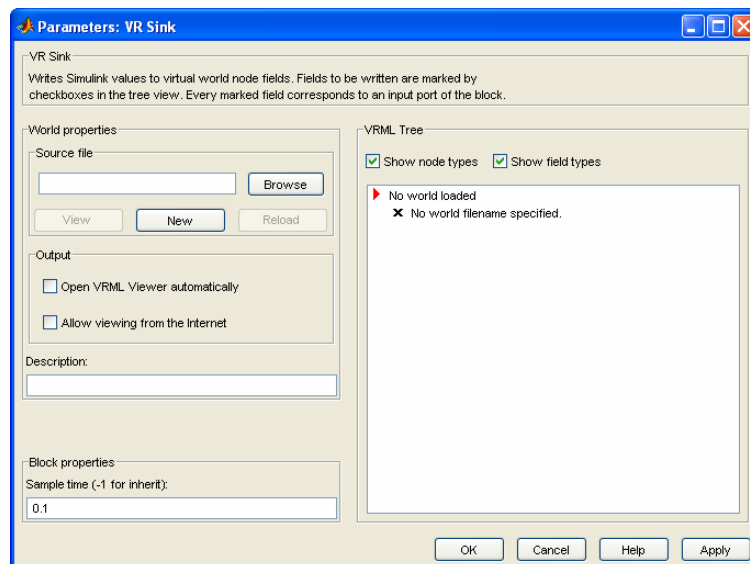


VR Sink

Účel Zapiše data z simulinku do virtuálního světa

Popis Blok VR Sink zapiše hodnotu z jeho portů do virtuálního světa specifikovaných v dialogovém okně Block Parameters.

Data VR Sink blok akceptuje všechna smysluplná data na vstupu, převede tyto datové typy do přirozených VRML typů.



Obr. 28 Dialogové okno Sink

Popis obr. 28:

Source file – Jméno souboru VRLM, které specifikuje virtuální svět, ke kterému je tento blok připojen. Tlačítko **View** umožňuje sledovat svět ve VR Toolbox prohlížeči nebo ve webovém prohlížeči. Tlačítko **Edit** spouští externí VRML editor a tlačítko **Reload** načte aktuální svět po provedení změn. Standardně se plná cesta k souvisejícímu *.wrl souboru objevuje v tomto text boxu. Pokud do tohoto pole vložíme pouze název souboru, VR Toolbox předpokládá, že se *.wrl soubor nachází ve stejném adresáři jako model file.

Open VRML viewer automatically – Pokud je zvolena tato možnost standardní VRML prohlížeč zobrazí virtuální svět po nahrání simulačního modelu.

Allow viewing from the Internet – Pokud je zvolena tato možnost, virtuální svět je možno prohlížet z klientského počítače. Pokud tato možnost není zvolena, svět je viditelný pouze na hostitelském počítači.

Descriptio – Popis, který je zobrazen ve všech výpisech virtuálních objektů, na popisové liště VR Toolboxu a v seznamu virtuálních světů na internetových stránkách VR Toolboxu.

Sample time – Zadává se čas ukázky

Poznámka : Pro lepší záznam animace je možné experimentálně měnit hodnotu této vlastnosti.

VRML Tree – Ukazuje strukturu VRML souboru a samotného virtuálního světa:

Uzly, které mají jméno, jsou označeny šipkami a jsou přístupné z Matlabu.

Uzly beze jmen, jejichž následníci jména mají, jsou označeny také červenými šipkami.

Toto označovací schéma umožňuje najít všechny přístupné uzly rozbalováním červených šipek. Ostatní uzly mají před svým jménem modrou tečku.

Pole s nastavitelnými hodnotami mají check boxy. Pomocí nich lze vybrat pole, do kterých Simulink vloží výstupní hodnoty (vstupní hodnoty pro virtuální svět). Pro každé vybrané pole je vytvořen v bloku vstupní port. Vstupní porty jsou přiřazeny k vybraným uzlům a polím, aby souhlasily s VRML souborem.

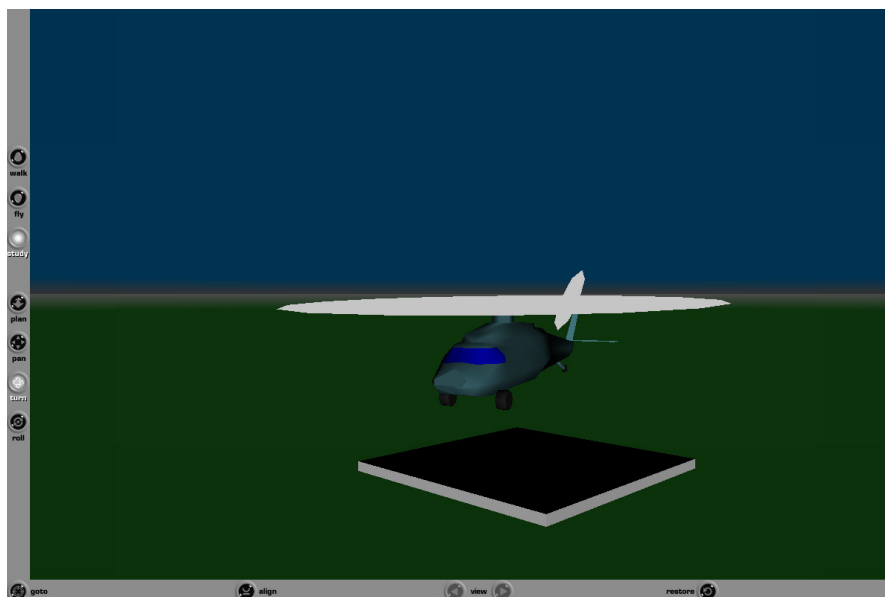
Pole, jejichž hodnota nemůže být nastavena (protože jejich nadřazené uzly nemají jména), mají ikonu ve tvaru x.

Show node types – Pokud je zvoleno, pak se ve VRML tree zobrazují typy uzlů.

Show field types – Pokud je zvoleno, pak se ve VRML tree zobrazují typy polí.

6.2 Propojení helikoptéry s VR Toolboxem

Na propojení použijeme Simulinkové schéma obr. 22. Samotná helikoptéra ve virtuálním světě je zobrazena na obr. 30.



Obr. 29 Helikoptéra zobrazena ve virtuálním světě

7 Závěr

Porovnáme-li regulátory Ziegler-Nicholsovou metodou (obr. 13), vyjde nám jako nejlepší regulátor PDf. U metody GMK vychází nejlepší regulátor také PDf a u frekvenčních metod vychází opět nejlepší PDf. Na obr. 21 je porovnání PDf regulátorů, zde vychází nejlépe regulátor navržený metodou GMK. Při praktickém ověřování se objevily další vlastnosti helikoptéry, které nejsou zahrnuty v rovnici (18). A to především, že model dokáže reagovat pouze na regulátory, které mají proporcionální složku min. 0,1. To znamená, že většina regulátorů, co jsme navrhli, nejde v praxi použít, neboť tuto podmínku nespĺňuje. Pokud je proporcionální složka kolem minimální hodnoty 0,1, můžeme pozorovat, že regulátor sice reaguje, ale regulace není příliš kvalitní (obr. 24). Optimální hodnota proporcionální složky je vidět na obr. 25 a je kolem 0,8. Nejlepší metodou pro návrh reálného regulátoru je v našem případě metoda cyklické variace konstant (obr. 25). To ale v praxi nemusí vždy vést k optimálnímu řešení. Nelze jednoznačně říct, který regulátor navržený metodou cyklické variace konstant je lepší, jestli PDf nebo PIDf. Záleží na konkrétní potřebě.

Pro využití Virtual Reality Toolboxu je nezbytné znát základy jazyka VRML, které jsou popsány v kapitole 5. Propojení našeho modelu s VR Toolboxem je na obr. 22. Zobrazení virtuálního světa je na obr. 30

8 Seznam použité literatury

- [1] Horáček P.: Systémy a modely. Skriptum FEL ČVUT 2001
- [2] John J.: Systémy a řízení. Skriptum FEL ČVUT 1998
- [3] Žilka J.: Diplomová práce – Simulace letu pomocí letového direktoru 2001, FEL ČVUT
- [4] Slavík A.: Diplomová práce – Trojrozměrné vizuální prostředí pro simulace, MFF UK 2003
- [5] Simulink (<http://mathworks.com>): Using Simulink 2006
- [6] Virtual Reality Toolbox (<http://mathworks.com>): User's Guide 2004
- [7] Učebnice SARI: (<http://dce.felk.cvut.cz/sari>) FEL ČVUT
- [8] Šebek M.: (<http://dce.felk.cvut.cz/sari>) Slides pro přednášky SRI FEL ČVUT 2005
- [9] Havel P.: (<http://dce.felk.cvut.cz/sari>) Frekvenční metody syntézy 2005, FEL ČVUT
- [10] <http://www.humusoft.cz/matlab/indexcz.htm>
- [11] VRML Tutoriál 2006:
(http://www.cgg.cvut.cz/vyuka/VRML/tutorial/pasmo/vrml_uvod_pasmo_1.html)
- [12] 3D VRML company: (<http://www.parallelgraphics.com>) 2006

9 Seznam použitých vývojových prostředků

Operační systémy

- MS Windows 2000 Profesional
- MS Windows 2003 Server standard
- MS Windows XP Profesional

Software

- Matlab 7.0.0 (R14)
 - Simulink 6.0 (R14)
 - Standardní Toolboxy Matlabu
 - Extended Real Toolbox 3.12
 - Virtual Reality Toolbox 4.0
- MS Office 2003
- V-Realm builder 2.0
- Cortona VRML Klient 4.2

10 Seznam obrázků

Obr. 1 Model helikoptéry (foto)	2
Obr. 2 Průběh helikoptéry na jednotkový skok	7
Obr. 3 Simulinkové schéma pro simulovaný systém	8
Obr. 4 Porovnání skutečného modelu a simulace	9
Obr. 5 Porovnání skutečného a identifikovaného modelu – upravené	9
Obr. 6 Bodeho charakteristika identifikovaného přenosu	10
Obr. 7 Nyquistova charakteristika identifikovaného přenosu	10
Obr. 8 Nicholsova charakteristika identifikovaného přenosu	11
Obr. 9 Určení kritické frekvence	12
Obr. 10 Simulinkové schéma pro měření regulátoru bez filtru	14
Obr. 11 Simulinkové schéma pro měření regulátoru s filtrem	14
Obr. 12 Porovnání regulátorů navržených Ziegler-Nicholsovou metodou	15
Obr. 13 Geometrické místo kořenů pro P regulátor	16
Obr. 14 Geometrické místo kořenů pro PI regulátor	16
Obr. 15 Geometrické místo kořenů pro PDf regulátor	17
Obr. 16 Geometrické místo kořenů pro PIDf regulátor	17
Obr. 17 Schéma použité v simulinku pro simulaci	18
Obr. 18 Porovnání regulátorů navržených GMK metodou	18
Obr. 19 Porovnání regulátorů navržených frekvenční metodou	21
Obr. 20 Porovnání PDf regulátorů	21
Obr. 21 Simulinkové schéma pro měření modelu	22
Obr. 22 PD regulátor navržený frekvenční metodou	22
Obr. 23 PID regulátor navržený frekvenční metodou	23
Obr. 24 Regulátory navržené metodou cyklické variace konstant	23
Obr. 25 Scéna zobrazená prohlížečem Cortona	25
Obr. 26 Dialogové okno Placeholder	29
Obr. 27 Dialogové okno Expander	29
Obr. 28 Dialogové okno Sink	30
Obr. 29 Helikoptéra zobrazena ve virtuálním světě	31