



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA
KYBERNETIKY

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

**Zpětnovazební řízení výtahů pro
kompenzaci nežádoucích vibrací**

Autor:
Břetislav Kubeš

Vedoucí práce:
Ing. Martin Gouběj, Ph.D.

Plzeň, 2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Břetislav KUBEŠ**
Osobní číslo: **A20B0331P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Automatické řízení a robotika**
Téma práce: **Zpětnovazební řízení výtahů pro kompenzaci nežádoucích vibrací**
Zadávací katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se s problematikou řízení pohybu výtahů. Sestavte matematický model dynamiky systému.
2. Navrhněte vhodnou strategii zpětnovazebního řízení pro kompenzaci nežádoucích vibrací kabiny v důsledku pružnosti závěsu a kompenzaci externích poruch. Zaměřte se na použití technik robustního řízení, gain-schedulingu, případně repetitive control a iterative learning control.
3. Otestujte navržené techniky řízení na simulačním modelu a diskutujte potenciální zlepšení v dosažené kvalitě řízení oproti čistě zpětnovazebnímu řešení.
4. V případě možnosti ověřte navržené řešení na vhodném mechatrickém systému emulujícím dynamiku výtahu.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

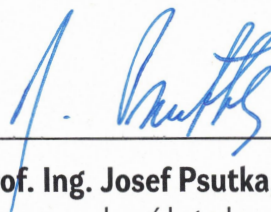
K.J. Astrom, R. M. Murray, Feedback systems, Princeton 2009 G. C. Goodwin, S. F. Graebe, M. E. Salgado, Control System Design, Pearson 2000 Materiály ke kurzům KKY/LS1, LS2, SM

Vedoucí bakalářské práce: **Ing. Martin Goubej, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **17. října 2022**
Termín odevzdání bakalářské práce: **22. května 2023**



Doc. Ing. Miloš Železný, Ph.D.
děkan



Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 17. října 2022

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 15. května 2023:

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Martinu Gubejovi, Ph.D. za cenné rady, vstřícnost a čas, který mi během vedení práce věnoval. Za rady chci poděkovat rovněž profesorovi Ing. Miloši Schlegelovi, CSc. A samozřejmě chci poděkovat i rodině za podporu při studiu.

Abstrakt

Cílem této práce je zamezit kmitání kabin výtahů, které zanáší do systému pružný mód v podobě pružnosti závěsného lana. V tom tkví i nelinearita systému - čím delší závěs je, tím více pruží. Práce vychází z již zpracovaného projektu [1], jehož zadáním bylo dostat se k takové metodě robustního řízení výtahů pro různá zatížení v různých patrech (rovnovážných bodech), která by dovolila co největší rychlost a utlumila kmity způsobené zrychlením. V této práci vlastnosti řízení ještě vylepšíme zavedením technik robustního zpětnovazebního řízení - gain-scheduling, linearizace podél trajektorie a řízení časově variantním LQ regulátorem (SDRE).

Navržené techniky otestujeme na simulačním modelu, a pokud bude možnost, tak i na vhodném mechatronickém systému emulujícím danou dynamiku.

Klíčová slova

výtah, zpětnovazební řízení, vibrace, gain scheduling, linearizace podél trajektorie, časově variantní LQR, SDRE

Abstract

This thesis aims to block elevator cabin vibrations caused by flexible mode of the system in the form of flexibility of suspension rope. This also causes the non-linearity of the system - the longer the rope, the more flexible it is. This piece of work extends our project [1] which aimed to work out the robust control method of elevators for different floors (equilibriums) and loads and which would allow us to achieve the highest possible speed while tempering oscillations caused by acceleration. In this work we will further improve the parameters of the regulation by robust feedback control methods, i.e. gain-scheduling, linearization around trajectory and time varying LQ regulator (SDRE).

These methods will then be tested on simulation and also on mechatronic system emulating lift dynamics if possible.

Keywords

elevator, lift, feedback control, vibrations, gain scheduling, linearization around trajectory, time varying LQR, SDRE

Obsah

Úvod	1
1 Popis systému	2
2 Rekapitulace projektu	3
2.1 Matematický model	3
2.2 Pohybové rovnice	3
2.3 Linearizace	4
2.4 Linearizace v různých výškách	7
2.5 Převodovka	9
2.6 Robustní regulace	12
2.6.1 Notch filtr	12
2.6.2 PI regulátor	14
2.7 Generátor trajektorie	19
2.8 Simulace robustní regulace na nelineárním modelu s generátorem trajektorie	21
2.9 Závěr projektu	24
3 Gain-scheduling	24
3.1 Selector	34
3.2 Proložení K_p a K_i křivkou	35
3.3 Dva regulátory	37
3.4 Odchylka polohy	38
4 Linearizace podél trajektorie	39
5 Časově proměnný lineárně kvadratický regulátor	48
6 Rekonstruktor	56
7 SDRE dynamický kompenzátor	63
8 Simulace s působením poruch	67
8.1 Porucha na kabině a_e	67
8.2 Vstupní porucha	74
9 Závěrečné srovnání zkoumaných strategií řízení	78
10 Repetitive control	81
11 Test na reálném systému	81
Závěr	82
Reference	83

Seznam obrázků

1	Úvodní obrázek	1
2	Zadaný systém	2
3	Špatný poměr rezonance a antirezonance	8
4	Zadaný systém	9
5	Bodeho charakteristiky přenosu na motor s převodovkou, parametry viz vztah 26, kabina v 1., 5. a 10. patře, $m_2 = 600 \text{ kg}$	11
6	Bodeho charakteristiky přenosu na motor s převodovkou, parametry viz vztah 26, kabina v 1. patře, $m_2 \in \{600, 1000, 1400\} \text{ kg}$	11
7	Bodeho charakteristika navrženého notch filtru	12
8	Porovnání Bodeho charakteristik přenosů s notch filtrem a bez	13
9	Schéma systému vhodného pro H_∞	14
10	Region všech stabilizujících reg. v aplikaci H_∞ , zbarvení podle ITAE kritéria	15
11	Přechodová charakteristika na straně kabiny, všech 9 systémů	16
12	Odezva na vstupní poruchu na straně kabiny, všech 9 systémů	16
13	Přechodová charakteristika na straně motoru, všech 9 systémů	17
14	Odezva na vstupní poruchu na straně motoru, všech 9 systémů	17
15	Bodeho charakteristika na straně kabiny v aplikaci H_∞ , všech 9 systémů	18
16	Znázornění přepínacích časů	19
17	Možnosti A nebo C1 a B nebo C2	20
18	Možnosti D1 a D2	20
19	Schéma modelu s generátorem trajektorie v Simulinku	21
20	Robustní regulace - rychlost motoru a detail (vpravo)	22
21	Robustní regulace - rychlosti a zrychlení	22
22	Robustní regulace - polohy kabiny a zátěže + detail (vpravo)	22
23	Robustní regulace - odchylky od generátoru	23
24	Robustní regulace - zrychlení kabiny pro $m_2 = 600 \text{ kg}$ a $m_2 = 1400 \text{ kg}$	23
25	Princip fungování gain-schedulingu	24
26	Porovnání přechodové charakteristiky mezi robustní regulací a GS	25
27	Porovnání odezvy na vstupní poruchu mezi robustní regulací a GS	25
28	Bodeho charakteristiky pro 1. patro	27
29	Nyquistova křivka pro 1. patro	27
30	Komplementární citlivostní funkce pro 1. patro	27
31	Bodeho charakteristiky pro 5. patro	28
32	Nyquistova křivka pro 5. patro	28
33	Komplementární citlivostní funkce pro 5. patro	28
34	Bodeho charakteristiky pro 10. patro	29
35	Nyquistova křivka pro 10. patro	29
36	Komplementární citlivostní funkce pro 10. patro	29
37	Simulační model pro gain-scheduling	30
38	Simulační model pro gain-scheduling - detail regulátoru	30
39	Gain-scheduling	31
40	Gain-scheduling - zrychlení kabiny	32
41	Gain-scheduling - změna parametrů regulátoru	32
42	Gain-scheduling - nespojitý výstup regulátoru	33
43	Gain-scheduling - nespojitý výstup regulátoru, detail	33
44	Problém s vysledováním u této implementace GS	34

45	Simulační model pro selectoring - detail regulátoru	35
46	Selectoring - zrychlení kabiny	35
47	Proložení parametrů křivkou - zrychlení kabiny	36
48	Proložení parametrů křivkou - změna parametrů regulátoru	36
49	Simulační model se dvěma regulátory - detail regulátoru	37
50	Dva regulátory - zrychlení kabiny	38
51	Odchylka polohy oproti GT při regulaci GS	38
52	Princip (časově proměnné) LQR regulace	49
53	Schéma SDRE regulace	49
54	Simulace SDRE stavové regulace, dolů	51
55	Simulace SDRE stavové regulace, nahoru	52
56	Schéma hybridní SDRE stavové - PI robustní regulace	53
57	Simulace hybridní SDRE stavové - PI robustní regulace, dolů	54
58	Simulace hybridní SDRE stavové - PI robustní regulace, nahoru	55
59	Princip fungování rekonstruktoru	56
60	Vnitřní schéma rekonstruktoru	57
61	Test ekvivalence systémů	60
62	Test ekvivalence systémů	60
63	Odhady rychlostí systému a odchylky od skutečných pro simulaci podél trajektorie	62
64	Schéma dynamického kompenzátoru	63
65	Rozdíly ve kvalitě rekonstrukce a z ní vyplývající LSZV	64
66	Implementace dynamického kompenzátoru v Simulinku	65
67	Simulace dynamického kompenzátoru	66
68	Simulace robustní regulace s poruchou $0.04ms^{-2}$ působící v čase $t = 15s$	67
69	Simulace robustní regulace s poruchou $0.6ms^{-2}$ působící v čase $t = 15s$	68
70	Simulace robustní regulace s poruchou $0.6ms^{-2}$ působící v čase $t = 20s$	68
71	Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 3m$	70
72	Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 15m$	70
73	Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 30m$	70
74	Impulsní charakteristiky - přenosy linearizované v $x_2 = 3m$	71
75	Impulsní charakteristiky - přenosy linearizované v $x_2 = 15m$	71
76	Impulsní charakteristiky - přenosy linearizované v $x_2 = 30m$	71
77	Simulace regulace dvěma regulátory s poruchou $0.6ms^{-2}$ působící v čase $t = 15s$	72
78	Simulace regulace DK s poruchou $0.6ms^{-2}$ působící v čase $t = 15s$	73
79	Schéma mimo systému použitého pro získání přenosů z působící chyby a_e	73
80	Simulace robustní regulace se vstupní poruchou	75
81	Simulace regulace dvěma regulátory se vstupní poruchou	76
82	Simulace dynamického kompenzátoru se vstupní poruchou	77
83	Rekonstrukce DK při simulaci se vstupní poruchou	78
84	Srovnání trajektorie s GT	79
85	Odchylky jednotlivých regulací oproti GT	80

Seznam významných vztahů

1	Vztahy pro délky lan	3
2	Výchozí pohybové rovnice systému	4
3	Moment potřebný k udržení rovnovážného stavu	4
4	Linearizovaný stavový model v obecné délce l_2	7
5	Spočtení konkrétních počátečních hodnot pro linearizaci	7
6	Virtuální změna fyzikálních parametrů	9
7	Fyzikální parametry modelu	10
8	Parametry notch filtru	12
9	Parametry regulátoru	14
10	Parametry simulace	21
11	Rovnice generátoru trajektorie	41
12	Parametry jednotlivých trajektorií	43
13	Linearizovaný systém s novými stavu torze δ_1, δ_2	59

Seznam značek

$\stackrel{!}{=}$ - musí být rovno
 \triangleq - je z definice rovno

Seznam zkratek

RB - rovnovážný(é) bod(y)
PI - proporcionálně-integrační
GT - generátor trajektorie
GS - gain-scheduling
DK - dynamický kompenzátor
LPT - linearizace podél trajektorie
IOE - vstupně-výstupně ekvivalentní (input/output equivalent)
LTV - lineární časově proměnný (linear time varying)
LQR - lineárně kvadratický regulátor
SDRE - časově proměnný LQR (State-Dependent Riccati Equation)
(L)SZV - (lineární) stavová zpětná vazba

G_m - bezpečnost v zesílení (gain margin)
 P_m - bezpečnost ve fázi (phase margin)
 S_m - bezpečnost ve stabilitě (stability margin)
 ω_B - šířka pásma (bandwidth frequency)
 S - citlivostní funkce uzavřené smyčky
 T - komplementární citlivostní funkce uzavřené smyčky
ISE kritérium . integrální kvadratické kritérium
(Integral of Square Error)
ITAE kritérium - integrální časově vážené kritérium z absolutní odchylky
(Integral of Time and Absolute Error)

Úvod

Výtahy jsou důležitý lidský vynález. Umožňují nám pohyb těžkých břemen a lidí do velkých výšek snadno a rychle využitím energie, původně mechanické, v dnešních dobách elektrické. Jelikož společnost a život v ní se neustále zrychlují, je kladen důraz i na stále vyšší rychlosti pohybu, a to nejen na silnici nebo železnici, ale i ve výtazích. A stejně jako vysokorychlostní železnice musí splňovat určité parametry komfortu pro cestující, i výtahy je nutno konstruovat a řídit tak, aby cestujícím bylo jejich využití příjemné, nebo alespoň nebylo nepříjemné.

Tak jako se na železnici mohou vyskytovat nerovnosti, které způsobí kymácení vlaku, tak i ve světě výtahů existuje podobný jev - pružení lana, na kterém visí kabina výtahu - který způsobí nepříjemné vibrace či dokonce zmítání kabiny.

Jelikož pružnost lana je přímo úměrná jeho délce, budou tyto nepříjemné vlastnosti tím horší, čím bude lano delší. V této práci zjistíme, že omezení kmitů zpětnovazebním řízením motoru (bez přídavných tlumičů, které využívají například výtahy Toshiba v mrakodrapu TAIPEI 101 [2] - nejrychlejší výtahy světa) je poměrně náročné i pro budovy běžných výšek (v kontextu České republiky), natož pro výškové budovy.

Z hlediska terminologie teorie řízení můžeme říci, že v pružnosti lan tkví hlavní nelinearita systému. Budeme se tedy pokoušet skrze linearizaci získat takový robustní regulátor motoru, který by řídil systém dostatečně robustně, aby byl aplikovatelný zpětně i na nelineární systém, a zároveň aby zachoval a dále vylepšil parametry jízdy, především omezil vibrace a oscilace způsobené rozjezdem a brzděním, ale i zrychlil samotný přesun z patra A do patra B.

Jelikož takovýto regulátor jsme získali již v projektu *KKY/PAŘR5: Robustní řízení nelineárního modelu výtahu ve vícero rovnovážných bodech* [1], ve spolupráci s Jakubem Tvrzem, budeme v této práci vycházet z výsledků projektu a nadále vylepšovat vlastnosti regulace využitím pokročilejších technik řízení, konkrétně metodami *gain-scheduling*, *linearizace podél trajektorie* a řízení *časově variantním LQ regulátorem*.



Obrázek 1: Úvodní obrázek

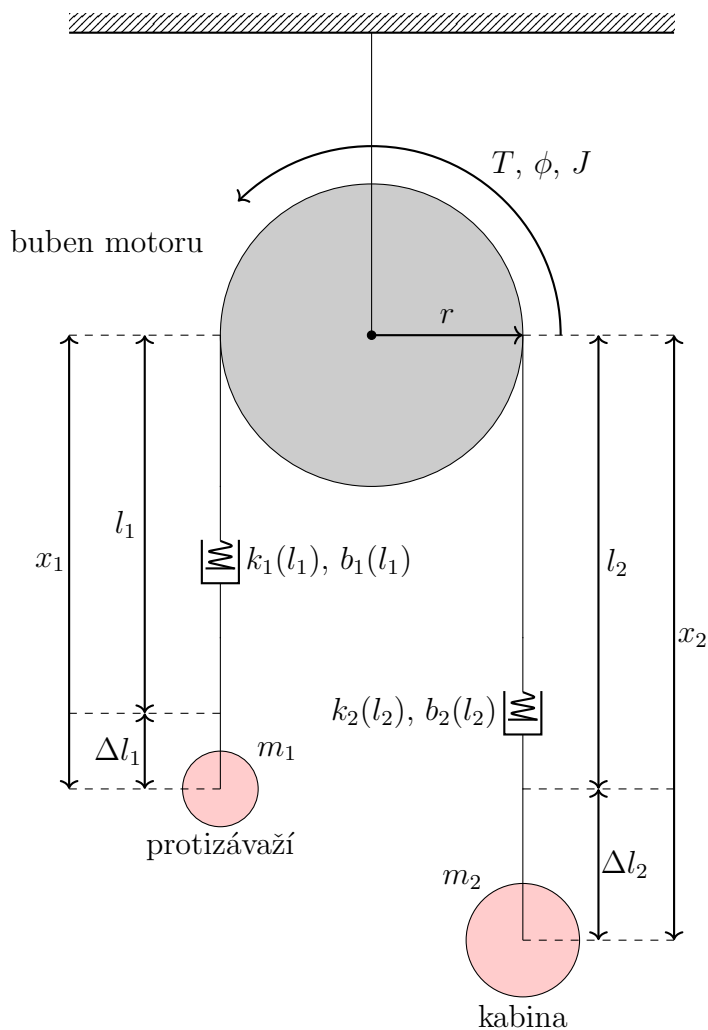
1 Popis systému

Základní systém budeme uvažovat následovně: Jedná se o motor, který dodává točivý moment T , na který je připevněna špulka (buben) s lanem, na jehož dvou koncích jsou připevněna kabina a protizávaží.

Špulka má určitý poloměr r a spolu s motorem mají moment setrvačnosti J .

Délka odtočeného lana na straně protizávaží v neprodlouženém stavu je pak rovna l_1 , přičemž kvůli pružnosti lana dojde k jeho prodloužení vlivem zatížení m_1 (dle Hookova zákona, více v sekci [Matematický model](#)), a tudíž vertikální vzdálenost od špulky motoru je $x_1 \neq l_1$. Pružnost lana na této straně závisí na jeho délce, tj. je jeho funkcí - $k_1(l_1)$, stejně tak tlumení $b_1(l_1)$.

Obdobně na straně kabiny, viz [obrázek 2](#) - $\{m_2, k_2(l_2), b_2(l_2)\} \Rightarrow x_2 \neq l_2$.



Obrázek 2: Zadaný systém

2 Rekapitulace projektu

2.1 Matematický model

Nejprve je třeba určit matematický popis modelu a získat vzhled do jeho dynamiky pomocí simulací. Získáme tak i dobrou představu o reálném systému a můžeme porovnat, zda se chová podle našich očekávání.

2.2 Pohybové rovnice

Při modelování musíme učinit některá zjednodušení, abychom nezískali příliš komplikovaný model, a proto budeme lana uvažovat jako systém pružina-tlumič. Při modelaci se inspirováme prací [3], která se mimo jiné zabývá modelováním obdobného systému, leč bez protizávaží.

Vyjdeme ze schématu - máme délky závěsných lan $l_{1/2}$, které uvažujeme v neprodlouženém stavu, a výšky zavěšených hmot $x_{1/2}$ (resp. jejich vzdálenost od bubnu motoru, tedy větší x znamená menší výšku), které reprezentují polohu hmot s již uvažovaným prodloužením dle Hookova zákona.

Vyjádříme si tedy délky lan:

$$l_2 = l_0 - \phi \cdot r ; l_1 = l_0 - l_2 - \pi r = \phi r - \pi r = (\phi - \pi)r \quad (1)$$

kde l_0 je klidová neprodloužená délka celého lana. Dále potřebujeme vyjádřit silové působení na jednotlivé hmoty (F_t - total force), kdy sílu lana působící na hmoty modelujeme jako systém pružina-tlumič ($F_{(s/d)i}$ - spring/damper force):

$$F_{t1} = m_1 g - F_{(s/d)1} - F_{f1} = m_1 g - k_1(x_1 - l_1) - b_1(\dot{x}_1 - \dot{l}_1) - F_{f1} \stackrel{!}{=} m_1 \ddot{x}_1 \quad (2)$$

$$F_{t2} = m_2 g - F_{(s/d)2} - F_{f2} = m_2 g - k_2(x_2 - l_2) - b_2(\dot{x}_2 - \dot{l}_2) - F_{f2} \stackrel{!}{=} m_2 \ddot{x}_2 \quad (3)$$

kde F_{fi} značí třecí sílu působící na hmoty v šachtě, ve které jsou usazené.

Též potřebujeme vyjádřit celkové silové působení na buben motoru (T_t - total torque):

$$T_t = T + F_{(s/d)1} \cdot r - F_{(s/d)2} \cdot r - T_f = \quad (4)$$

$$= T + k_1(x_1 - l_1)r + b_1(\dot{x}_1 - \dot{l}_1)r - k_2(x_2 - l_2)r - b_2(\dot{x}_2 - \dot{l}_2)r - T_f \stackrel{!}{=} J \cdot \ddot{\phi}$$

což je logické, jelikož točivý moment $T \triangleq F \cdot r$.

T_f zde reprezentuje tření uvnitř motoru na bubnu - $T_f = b_f \cdot \omega = b_f \cdot \dot{\phi}$, T je pak vstupní točivý moment udělovaný motorem.

Dále budeme uvažovat, že hodnoty k , b systému pružina-tlumič jsou proměnné a závisí na délce příslušného lana:

$$k_1(l_1) = \frac{k_0}{l_1}, b_1(l_1) = \frac{b_0}{l_1}, k_2(l_2) = \frac{k_0}{l_2}, b_2(l_2) = \frac{b_0}{l_2} \quad (5)$$

kde k_0 a b_0 jsou jednotková tuhost pružiny a jednotkový součinitel útlumu.

Toto bude posléze způsobovat, že módy systému (a tedy i vibrace kabiny) budou časově proměnné a bude tedy nutno tomuto přizpůsobit i řízení.

První dvě rovnice vydělíme příslušnou hmotností (k osvobození nejvyšší derivace) a dosadíme za l_1 a l_2 (pozor na derivace konstant - např. πr - jsou nulové, proto v následujících rovnicích již nejsou) a také za k_1, b_1, k_2, b_2 .

Dostaneme tedy následující rovnice:

$$\ddot{x}_1 = g - \frac{k_0}{m_1(\phi r - \pi r)}(x_1 - \phi r + \pi r) - \frac{b_0}{m_1(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) - \frac{F_{f1}}{m_1} \quad (6)$$

$$\ddot{x}_2 = g - \frac{k_0}{m_2(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0}{m_2(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - \frac{F_{f2}}{m_2} \quad (7)$$

$$\begin{aligned} \ddot{\phi} = & \frac{T}{J} + \frac{k_0 r}{J(\phi r - \pi r)}(x_1 - \phi r + \pi r) + \frac{b_0 r}{J(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) + \dots \\ & \dots - \frac{k_0 r}{J(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0 r}{J(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - b_f \frac{\dot{\phi}}{J} \end{aligned} \quad (8)$$

2.3 Linearizace

Linearizace probíhá vždy v rovnovážném bodě (alternativně podél pracovní trajektorie). V něm jsou rychlosti nulové a polohy konstantní, tudíž všechny derivace nabývají nulové hodnoty. Vyjádříme si tedy jednotlivé proměnné (x_1 , x_2 , ϕ) čímž dostaneme požadované počáteční podmínky v RB a rovněž moment síly T (resp. točivý moment), který potřebujeme k udržení soustavy v RB.

Jelikož chceme získat obecný lineární model pro obecnou délku lana l , vyjádříme nakonec výrazy pomocí proměnných l_1 a l_2 (viz [rovnice 1](#)).

První rovnice ([rovnice 6](#))

Derivace vynulujeme a vyjádříme x_1 , čímž dostaneme první počáteční podmínku:

$$x_{10} = \frac{l_1(gm_1 - F_{f1} + k_0)}{k_0} \quad (9)$$

Druhá rovnice ([rovnice 7](#))

Derivace vynulujeme a vyjádříme x_2 , čímž dostaneme druhou počáteční podmínku:

$$x_{20} = \frac{l_2(gm_2 - F_{f2} + k_0)}{k_0} \quad (10)$$

Třetí rovnice ([rovnice 8](#))

Derivace vynulujeme a dosadíme již vyjádřené x_{20} a x_{10} . Získáme tak moment potřebný pro udržení rovnovážného stavu.

$$T = gr(m_2 - m_1) + r(F_{f1} - F_{f2}) \quad (11)$$

Počáteční podmínka pro ϕ_0 bude logicky dána jen délkami l_2 a l_1 , jedná se totiž o natočení bubnu motoru.

Budeme předpokládat, že volíme délku lana l_2 , jelikož to je délka lana na straně kabiny. Pak pro l_1 platí [vztah 1](#) a z ní odvodíme počáteční hodnotu ϕ_0 (v radiánech):

$$\phi_0 = \frac{l_1 + \pi r}{r} \quad (12)$$

Ještě si zavedeme následující stavové proměnné:

$$\begin{aligned} [z_1, z_2, z_3, z_4, z_5, z_6] &= [x_1, x_2, \phi, \dot{x}_1, \dot{x}_2, \dot{\phi}] \\ \dot{z}_1 &= z_4 \\ \dot{z}_2 &= z_5 \\ \dot{z}_3 &= z_6 \\ \dot{z}_4 &= g - \frac{k_0}{m_1(z_3 r - \pi r)}(z_1 - z_3 r + \pi r) - \frac{b_0}{m_1(z_3 r - \pi r)}(z_4 - z_6 r) - \frac{F_{f1}}{m_1} \\ \dot{z}_5 &= g - \frac{k_0}{m_2(l_0 - z_3 r)}(z_2 - l_0 + z_3 r) - \frac{b_0}{m_2(l_0 - z_3 r)}(z_5 + z_6 r) - \frac{F_{f2}}{m_2} \\ \dot{z}_6 &= \frac{T}{J} + \frac{k_0 r}{J(z_3 r - \pi r)}(z_1 - z_3 r + \pi r) + \frac{b_0 r}{J(z_3 r - \pi r)}(z_4 - z_6 r) + \dots \\ &\dots - \frac{k_0 r}{J(l_0 - z_3 r)}(z_2 - l_0 + z_3 r) - \frac{b_0 r}{J(l_0 - z_3 r)}(z_5 + z_6 r) - b_f \frac{z_6}{J} \end{aligned} \quad (13)$$

Nyní můžeme začít linearizovat v okolí tohoto RB (v rovnicích můžeme za T dosadit [vztah 11](#)). Derivujeme podle příslušných proměnných, abychom naplnili matici dynamiky stavového modelu (**A**).

Parciální derivace prvních tří rovnic jsou zjevné, dalších tří nikoli:

Parciální derivace \dot{z}_4 ; uvažujeme $z_3 r - \pi r = l_1$ ([vztah 1](#), $z_3 = \phi$).

$$\begin{aligned} \frac{\partial \dot{z}_4}{\partial z_1} &= \frac{-k_0}{m_1 l_1} \\ \frac{\partial \dot{z}_4}{\partial z_2} &= 0 \\ \frac{\partial \dot{z}_4}{\partial z_3} &= \frac{k_0 r z_1}{m_1 l_1^2} \\ \frac{\partial \dot{z}_4}{\partial z_4} &= \frac{-b_0}{m_1 l_1} \\ \frac{\partial \dot{z}_4}{\partial z_5} &= 0 \\ \frac{\partial \dot{z}_4}{\partial z_6} &= \frac{b_0 r}{m_1 l_1} \end{aligned} \quad (14)$$

Parciální derivace \dot{z}_5

$$\begin{aligned}
\frac{\partial \dot{z}_5}{\partial z_1} &= 0 \\
\frac{\partial \dot{z}_5}{\partial z_2} &= \frac{-k_0}{m_2 l_2} \\
\frac{\partial \dot{z}_5}{\partial z_3} &= \frac{-k_0 r z_2}{m_2 l_2^2} \\
\frac{\partial \dot{z}_5}{\partial z_4} &= 0 \\
\frac{\partial \dot{z}_5}{\partial z_5} &= \frac{-b_0}{m_2 l_2} \\
\frac{\partial \dot{z}_5}{\partial z_6} &= \frac{-b_0 r}{m_2 l_2}
\end{aligned} \tag{15}$$

Parciální derivace \dot{z}_6

$$\begin{aligned}
\frac{\partial \dot{z}_6}{\partial z_1} &= \frac{k_0 r}{J l_1} \\
\frac{\partial \dot{z}_6}{\partial z_2} &= \frac{-k_0 r}{J l_2} \\
\frac{\partial \dot{z}_6}{\partial z_3} &= \frac{-k_0 r^2 (z_2 l_1^2 + z_1 l_2^2)}{J l_1^2 l_2^2} \\
\frac{\partial \dot{z}_6}{\partial z_4} &= \frac{b_0 r}{J l_1} \\
\frac{\partial \dot{z}_6}{\partial z_5} &= \frac{-b_0 r}{J l_2} \\
\frac{\partial \dot{z}_6}{\partial z_6} &= \frac{-r^2 (l_1 + l_2) b_0 - b_f l_1 l_2}{J l_1 l_2}
\end{aligned} \tag{16}$$

Ve výrazech se stále vyskytují stavové proměnné z_i . To je v pořádku, jelikož linearizujeme pro obecnou délku závěsu l_2 . Po zvolení konkrétní délky $l_2 = l_{20}$ (resp. budeme volit spíše reálnou výšku po prodloužení $x_2 = x_{20}$ a z ní l_{20} vypočteme), ve které budeme linearizovat, tyto hodnoty do výrazů dosadíme a získáme výrazy již bez stavových proměnných.

Matici vstupu (\mathbf{B}) budou tvořit parciální derivace vstupu (tedy momentu), který se vykytuje jen v poslední rovnici:

$$\frac{\partial \dot{z}_6}{\partial T} = \frac{1}{J} \tag{17}$$

Výstupní matici \mathbf{C} naplníme jedničkami na pozicích stavů, které chceme pozorovat, všude jinde budou nuly. Například tak, jak je to v následujícím případě, kde nás zajímají rychlost kabiny \dot{x}_2 a rychlost otáčení motoru $\dot{\phi}$.

Stavy označujeme jako \mathbf{z} proto, že standardně využívané \mathbf{x} již využíváme pro výšky kabiny (x_2) a protizátěže (x_1).

Vstupem je moment motoru, čili $\mathbf{u} = T$.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_0}{m_1 l_1} & 0 & \frac{k_0 r z_1}{m_1 l_1^2} & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \frac{-k_0}{m_2 l_2} & \frac{-k_0 r z_2}{m_2 l_2^2} & 0 & \frac{-b_0}{m_2 l_2} & \frac{-b_0 r}{m_2 l_2} \\ \frac{k_0 r}{J l_1} & \frac{-k_0 r}{J l_2} & \frac{-k_0 r^2 (z_2 l_1^2 + z_1 l_2^2)}{J l_1^2 l_2^2} & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2 (l_1 + l_2) b_0 - b_f l_1 l_2}{J l_1 l_2} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{J} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t) \quad (18)$$

2.4 Linearizace v různých výškách

Již máme vyjádřený obecný lineární model. Není tudíž problém dosadit hodnotu x_{20} (skutečná počáteční poloha kabiny) v místech, kde předpokládáme jednotlivá patra. Vztahy pro počáteční podmínky pak vypadají následovně:

$$x_{20} = \text{zvolíme}$$

$$l_{20} = \frac{x_{20} k_0}{g \cdot m_2 - F_{f1} + k_0} \quad (19)$$

$$l_{10} = l_0 - \pi r - l_{20} \quad (20)$$

$$x_{10} = \frac{l_{10} (g \cdot m_1 - F_{f1} + k_0)}{k_0} \quad (21)$$

$$\phi_0 = \frac{l_{10} + \pi r}{r} \quad (22)$$

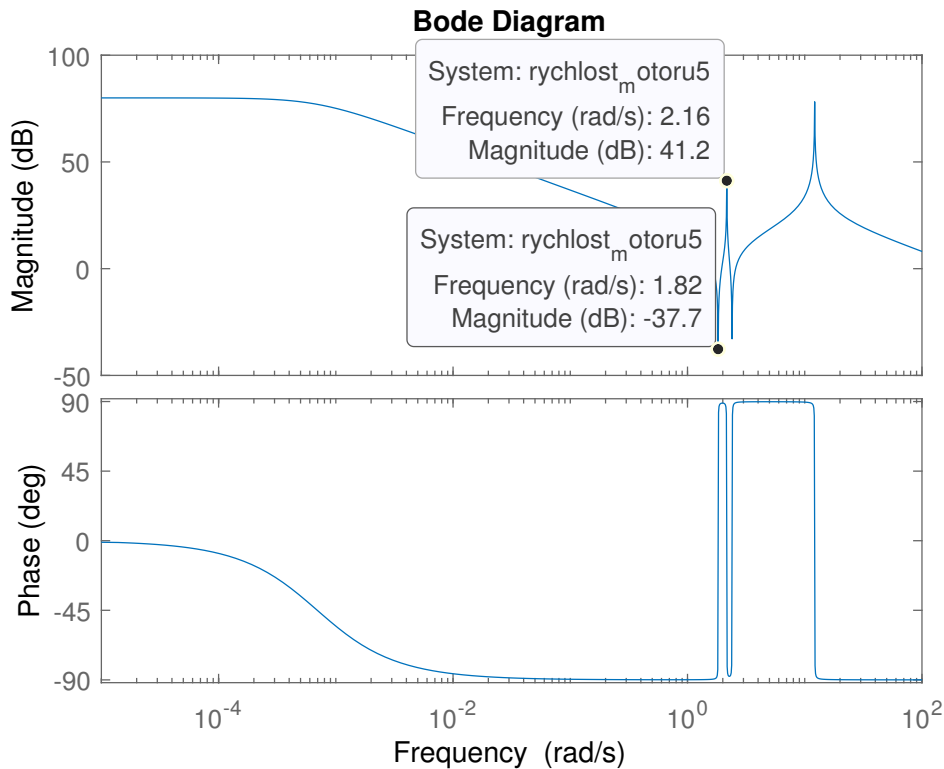
Když tyto hodnoty dosadíme do obecného linearizovaného modelu (viz [vztah 18](#)), dostaneme stavový model linearizovaný v konkrétním patře.

Budeme nyní uvažovat 10 patrovou budovu, tudíž získáme 10 stavových modelů. Z těch není těžké vyjádřit přenosy, které potřebujeme, a to přenos na rychlost motoru ($\dot{\phi}$) a na rychlost kabiny (\dot{x}_2) ze vstupního momentu T . Rychlost motoru potřebujeme z toho důvodu, že polohu kabiny budeme řídit nepřímo právě pomocí rychlosti motoru.

Po vykreslení Bodeho charakteristiky přenosu na rychlost motoru (například v pátém patře) však dojdeme k nepříjemnému zjištění. Ukazuje se totiž, že aby byl systém dobře říditelný, je potřeba, aby byl poměr frekvencí první antirezonance a rezonance ideálně roven cca 2, respektive $\in \langle \sqrt{3}, \sqrt{5} \rangle$ [4, p. 832]. U nás je bohužel roven zhruba 1.187 (viz obrázek 3), tz. systém je velmi špatně říditelný.

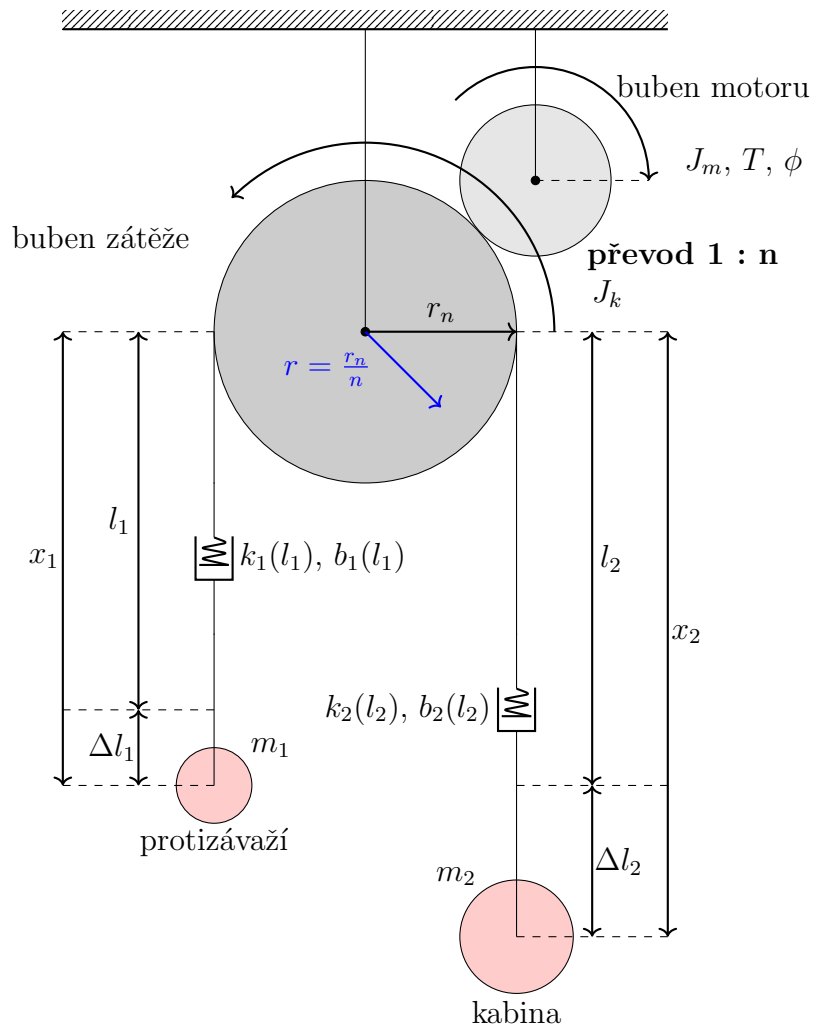
Z toho důvodu musíme do modelu zařadit ještě převodovku. Ta by byla v realitě stejně potřeba, jelikož výtah se pohybuje relativně pomalu, ovšem pro elektrický motor jsou ideální mnohem vyšší pracovní otáčky.

Zavedením převodu rovněž ovlivníme fyzikální parametry (například virtuálně změníme poloměr špulky motoru a moment setrvačnosti), což by mohlo dopomoci k lepší říditelnosti.



Obrázek 3: Špatný poměr rezonance a antirezonance

2.5 Převodovka



Obrázek 4: Zadaný systém

Jak jsme již zmínili, pro pohybové rovnice tohoto systému dojde k virtuální změně poloměru špulky a jejího momentu setrvačnosti:

$$J = J_m + \frac{J_k}{n^2}$$

$$r = \frac{r_n}{n} \quad (23)$$

Kdy J_m značí moment setrvačnosti motoru a J_k moment setrvačnosti kladky (buben zátěže). Kladku modelujeme zjednodušeně pomocí válce, tudíž dostáváme:

$$J_k = \frac{1}{2} m_k r_n^2 \quad (24)$$

Moment setrvačnosti motoru definujeme na základě přibližně reálných parametrů [3] následovně:

$$J_m = \frac{1}{2} \cdot 0,5 \cdot 0,02^2 \quad (25)$$

Pokud tyto vztahy dosadíme do původních rovnic (viz vztahy 6 až 8), získáme rovnice nové, a to včetně libovolného převodu na straně motoru.

Zvolené parametry by měly relativně dobře odpovídat realitě (vztah 26). Znovu vykreslená Bodeho charakteristika s těmito parametry (viz obrázek 5 - průběhy v 1., 5. a 10. patře) již vypadá lépe, píky rezonance a antirezonance nejsou tak ostré.

převodový poměr: $n = 15$

tíhové zrychlení: $g = 9.81 \text{ [m} \cdot \text{s}^{-2}\text{]}$

hmotnost zátěže: $m_1 = 1000 \text{ [kg]}$

hmotnost kabiny: $m_2 = 600 - 1400 \text{ [kg]}$

hmotnost kladky: $m_k = 50 \text{ [kg]}$

celková délka lana: $l_0 = 33 \text{ [m]}$

jednotková tuhost pružiny: $k_0 = 5000000 \text{ [N]}^{[1]}$

jednotkový součinitel útlumu: $b_0 = 60000 \text{ [N} \cdot \text{s]}^{[1]}$

skutečný poloměr špulky motoru: $r_n = 0.3 \text{ [m]}$

moment setrvačnosti motoru: $J_m = 1/2 \cdot 0.5 \cdot 0.02^2 \text{ [kg} \cdot \text{m}^2\text{]}$

moment setrvačnosti kladky: $J_k = 1/2 \cdot m_k \cdot r_n^2 \text{ [kg} \cdot \text{m}^2\text{]}$

virtuální přeškálovaný poloměr: $r = r_n/n$

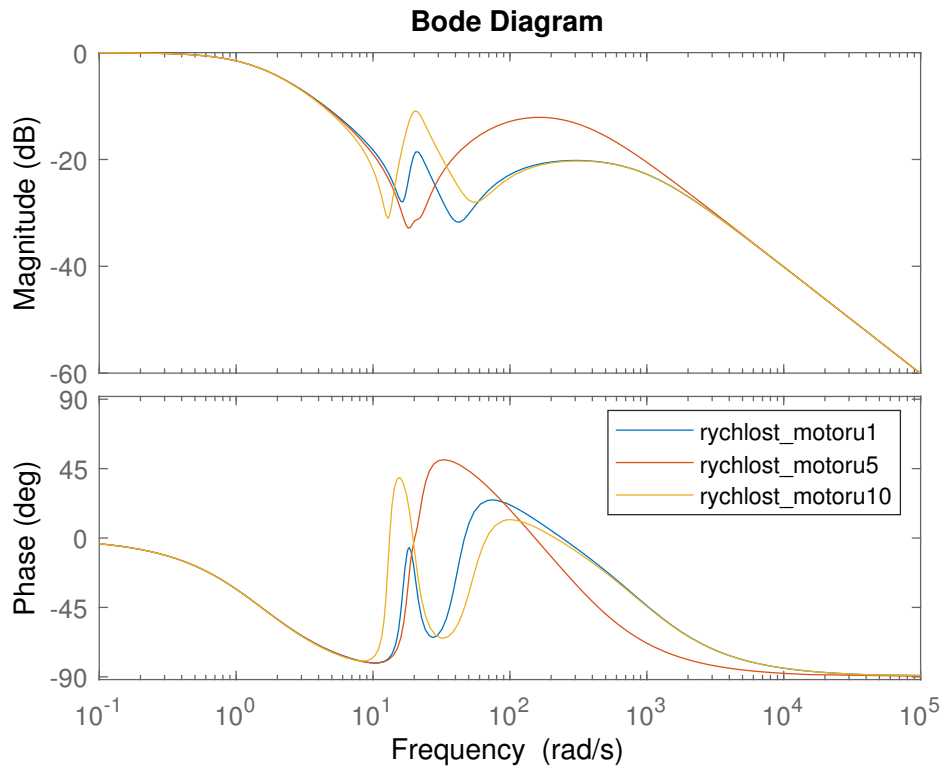
virtuální moment setrvačnosti: $J = J_m + J_k/n^2 \quad (26)$

Ani přidání převodovky však nestačilo k návrhu dostatečně robustního regulátoru. Velký problém představuje především proměnlivá hmotnost kabiny (viz obrázek 6), která pohybuje s frekvencí první rezonance.

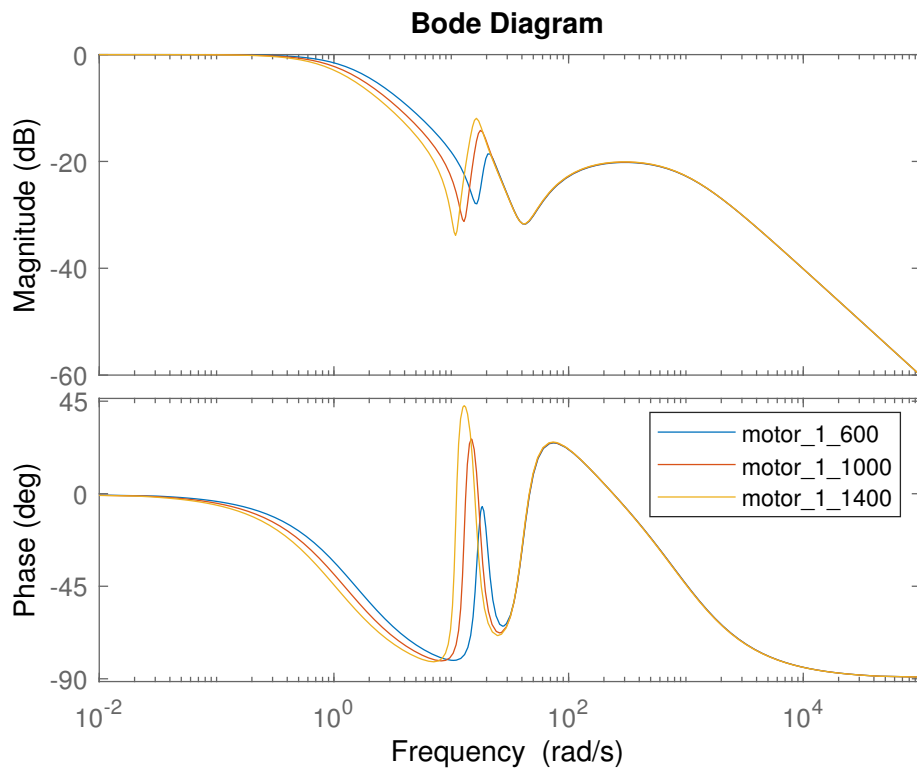
V kombinaci s různými patry, které mění zase velikost rezonance (amplitudu), dostáváme příliš širokou škálu možností, kde a jaká rezonance bude, což nám zabraňuje v návrhu robustního regulátoru.

Proto je další část věnována návrhu notch filtru.

^[1]Protože jednotkou tuhosti k je $N \cdot m^{-1}$ a jednotkou délky l je metr, musí mít jednotková tuhost pružiny k_0 za svou jednotku Newtony. Podobně součinitel útlumu b má jednotku $N \cdot s \cdot m^{-1}$ a tak jednotkou b_0 musí být $N \cdot s$



Obrázek 5: Bodeho charakteristiky přenosu na motor s převodovkou, parametry viz [vztah 26](#), kabina v 1., 5. a 10. patře, $m_2 = 600 \text{ kg}$



Obrázek 6: Bodeho charakteristiky přenosu na motor s převodovkou, parametry viz [vztah 26](#), kabina v 1. patře, $m_2 \in \{600, 1000, 1400\} \text{ kg}$

2.6 Robustní regulace

2.6.1 Notch filtr

Základní myšlenkou vedoucí k návrhu notch filtru (pásmová zadrž) byla potřeba zbavit se první rezonance. Notch filtr se má stát vlastně součástí regulátoru, což povede k jisté nevýhodě - jestliže určité frekvence takovýmto způsobem zatlumíme, bude horší reakce regulátoru na poruchy na těchto frekvencích.

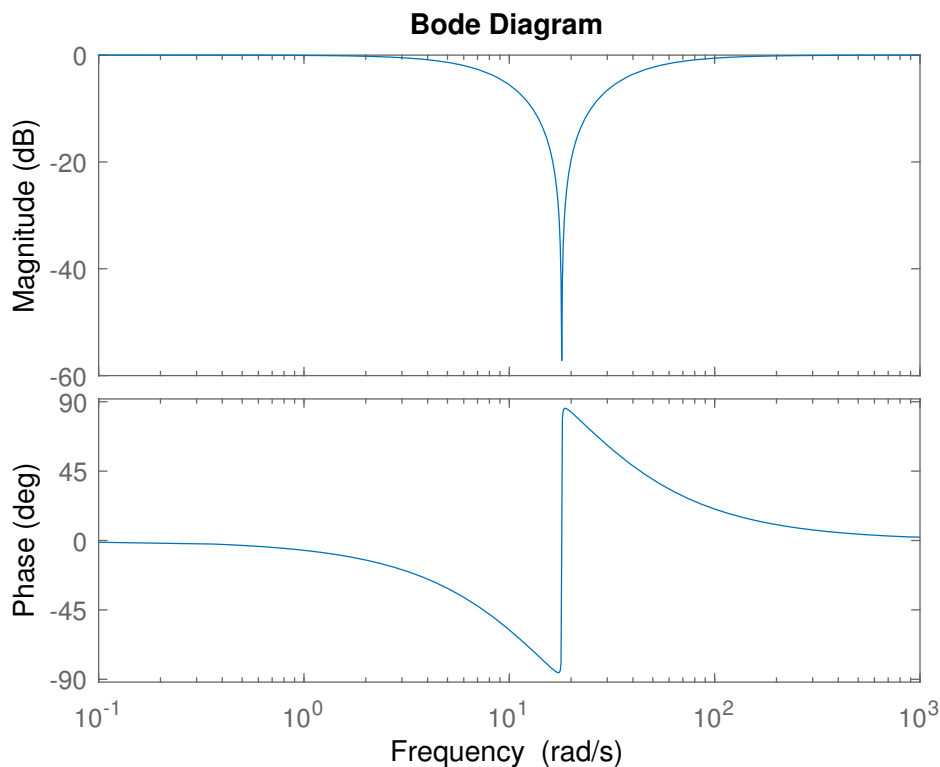
Notch filtr je možné reprezentovat přenosovou funkcí ve tvaru:

$$Notch(s) = \frac{s^2 + 2 \cdot \xi_1 \omega_n + \omega_n^2}{s^2 + 2 \cdot \xi_2 \omega_n + \omega_n^2} \quad (27)$$

kde parametr ω_n určuje frekvenci, na které se bude nacházet pík filtru, a parametr ξ_2 určuje jeho šířku. Parametr ξ_1 pak určuje jeho hloubku. Tento filtr (obrázek 7) bylo třeba navrhnout tak, aby dokázal odfiltrovat první rezonanci u všech přenosů, tudíž s dostatečnou (ale ne přílišnou) šířkou.

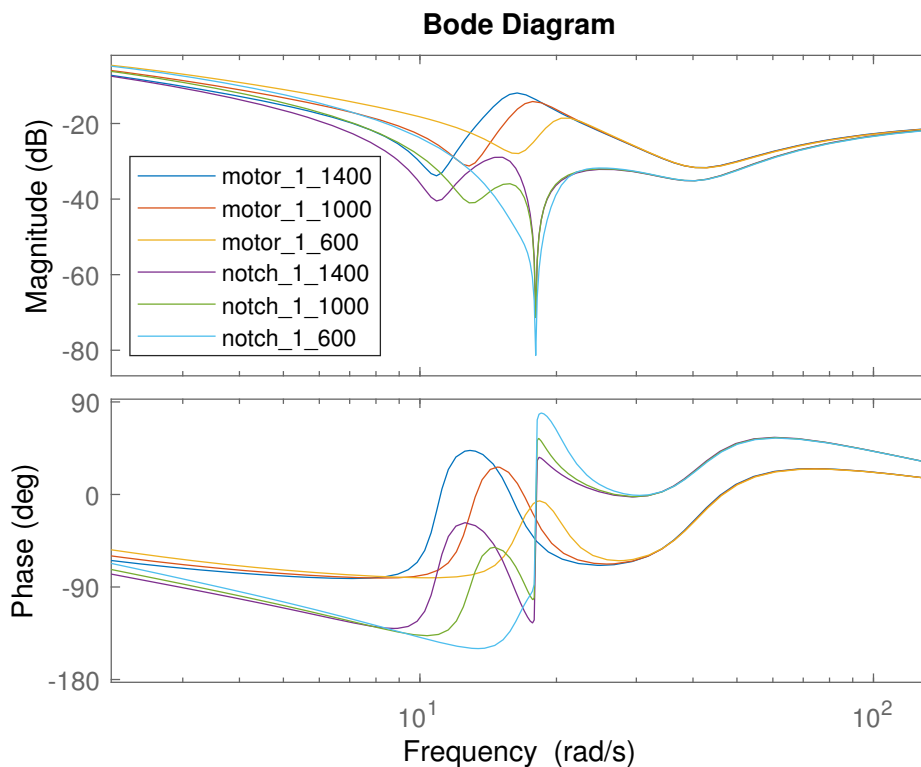
Parametry námi naladěného notch filtru vypadají následovně (vztah 28):

$$\begin{aligned} \omega_n &= 18 \\ \xi_1 &= 1 \\ \xi_2 &= \frac{1}{720} \end{aligned} \quad (28)$$



Obrázek 7: Bodeho charakteristika navrženého notch filtru

Na obrázku 8 lze vidět výsledek, když filtr aplikujeme na přenosy:

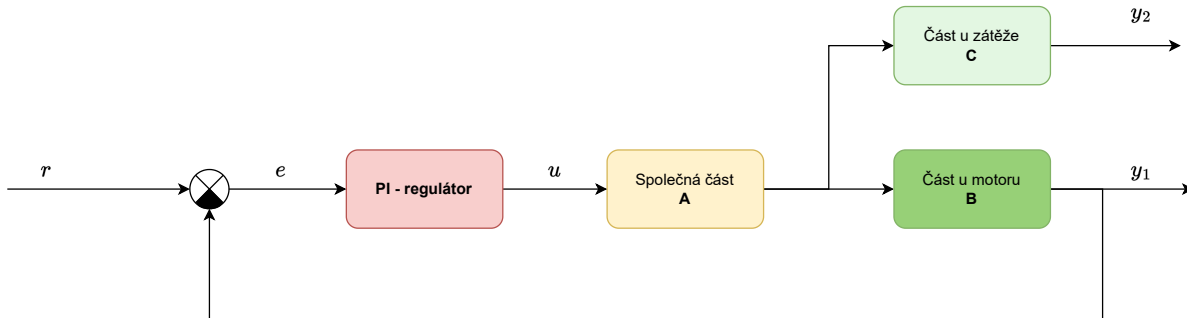


Obrázek 8: Porovnání Bodeho charakteristik přenosů s notch filtrem a bez

Skutečně jsme rezonance vhodně potlačili. Dále bude potřeba zahrnout tento filtr do regulátoru (zřejmě PI regulátor) který potřebujeme navrhnout tak, aby fungoval pro všechny přenosy - tedy jak v různých patrech tak i pro různou míru zatížení.

2.6.2 PI regulátor

Pro tvorbu regulátoru jsme se rozhodli využít Matlab aplikaci H_∞ z dílen KKY FAV ZČU (autorem Michal Špírk [5]), jelikož je zamýšlený přesně pro tento typ úloh - řízení systému zpětnou vazbou z jedné části smyčky pro dosažení kvality řízení v jiné části smyčky.[6]



Obrázek 9: Schéma systému vhodného pro H_∞ [6, p. 7]

Z obrázku 9 vidíme, že systém má jednu společnou část a pak dvě oddělené části. Společnou částí **A** pro nás bude pouze notch filtr, viz sekce 2.6.1, oddělenou částí **B** je přenos z momentu na rychlost otáčení motoru $T \rightarrow \dot{\phi}$ a oddělenou částí **C** je přenos z momentu na rychlost kabiny $T \rightarrow \dot{x}_2$ (přes uvažovanou převodovku, viz obrázek 4).

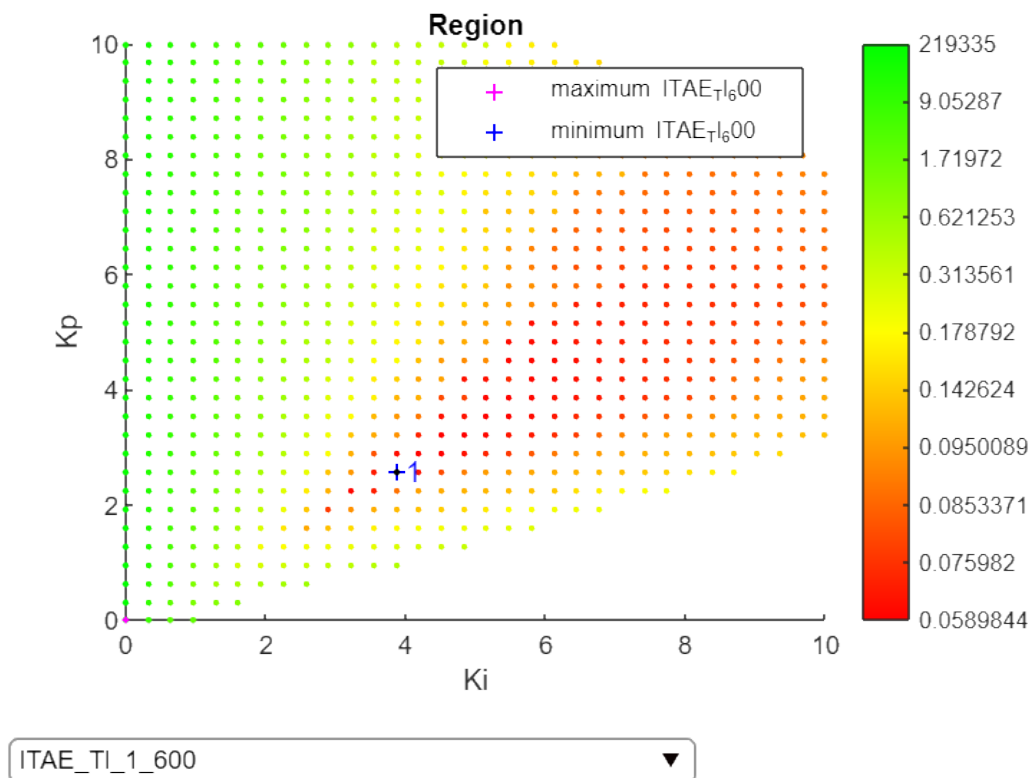
Jelikož aplikace vypočítává vhodný regulátor stabilizující všechny zadané přenosy (metodou H-nekonečno), zadáme do ní celkem 18 přenosů - 3 různá patra (1., 5. a 10.) krát 3 různá zatížení (600, 1000 a 1400 kg) krát 2 (přenos na buben zátěže a na kabinu) - plus 9x notch filtr (ke každé dvojici jeden, ten samý, viz sekce 2.6.1). Tím dosáhneme kýženého efektu, tedy že regulátor bude stabilizovat pro malá i velká zatížení v nejnižším i nejvyšším patře.

Aplikace nám nedá ale pouze jeden regulátor, dá nám jich celou množinu. Všechny jsou stabilizující, ovšem jsou uspořádány podle nějakého kritéria. Aplikace nám umožňuje vybrat mnoho kritérií - my jsme vybrali ITAE na straně zátěže (tz. pro kabinu), jelikož jej považujeme za nejvýhodnější pro náš záměr (dosažení co nejmenších kmitů co nejrychleji), a to v nejnižším patře s nejmenším zatížením, jelikož tento systém nejvíce kmitá a je ze všech nejhůře říditelný.

Vybereme tedy regulátor s nejnižším ITAE kritériem (menší = lepší), viz obrázek 10, a dostaneme kýžený regulátor:

$$F_R = \frac{2.581s + 3.871}{s} \quad (29)$$

Tz. dostaneme PI regulátor s $K_i = 3.871$ a $K_p = 2.581$.



Obrázek 10: Region všech stabilizujících reg. v aplikaci H_∞ , zbarvení podle ITAE kritéria

Na [obrázku 11](#) můžeme vidět přechodové charakteristiky všech devíti systémů (na straně kabiny) s tímto regulátorem ([vztah 29](#)).

Na [obrázku 12](#) pak vidíme odezvu na vstupní poruchu na straně kabiny, která je o něco pomalejší a má ne příliš pěkný průběh, což je dáno právě tím, že použitý notch filtr kazí odregulování poruch právě na zatlumených frekvencích.

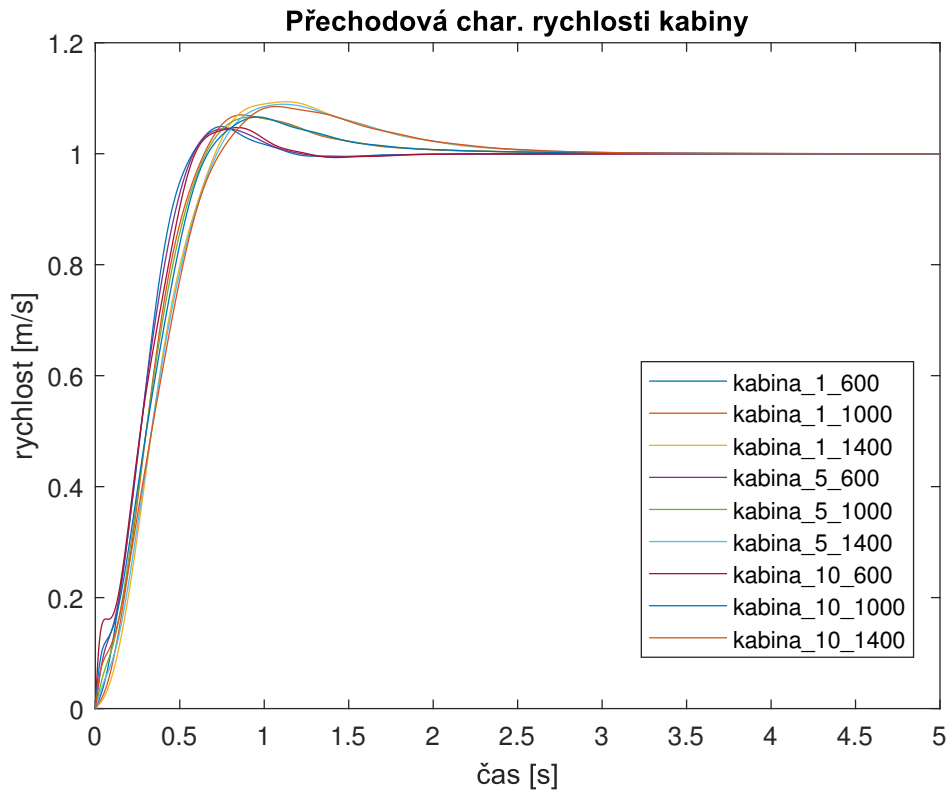
Na [obrázku 13](#) jsou znázorněné přechodové charakteristiky všech devíti systémů (na straně motoru).

Na [obrázku 14](#) je ještě odezva na vstupní poruchu na straně motoru.

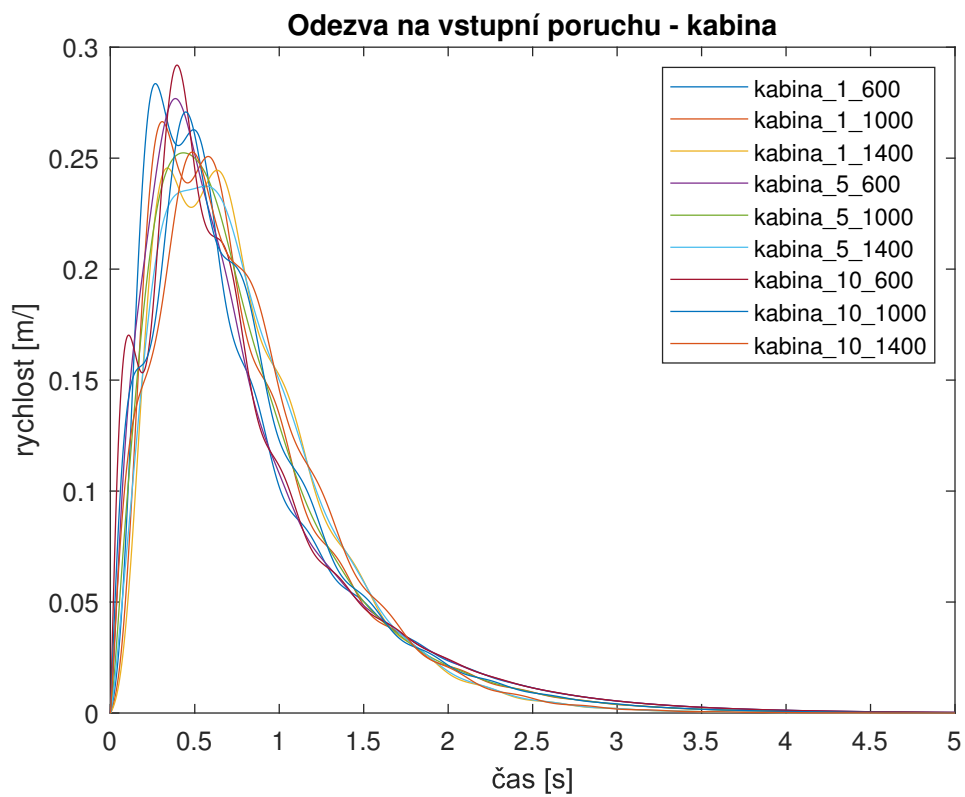
Na [obrázku 15](#) jsou pak vyobrazeny Bodeho charakteristiky všech devíti systémů (na straně kabiny) s tímto regulátorem.

Získali jsme tedy vhodný robustní regulátor, který dokáže řídit náš výtah v různých patrech pro různá zatížení a to s poměrně dobrými vlastnostmi (nízký překmit, rychlá regulace chyby, malá kmitavost regulátoru).

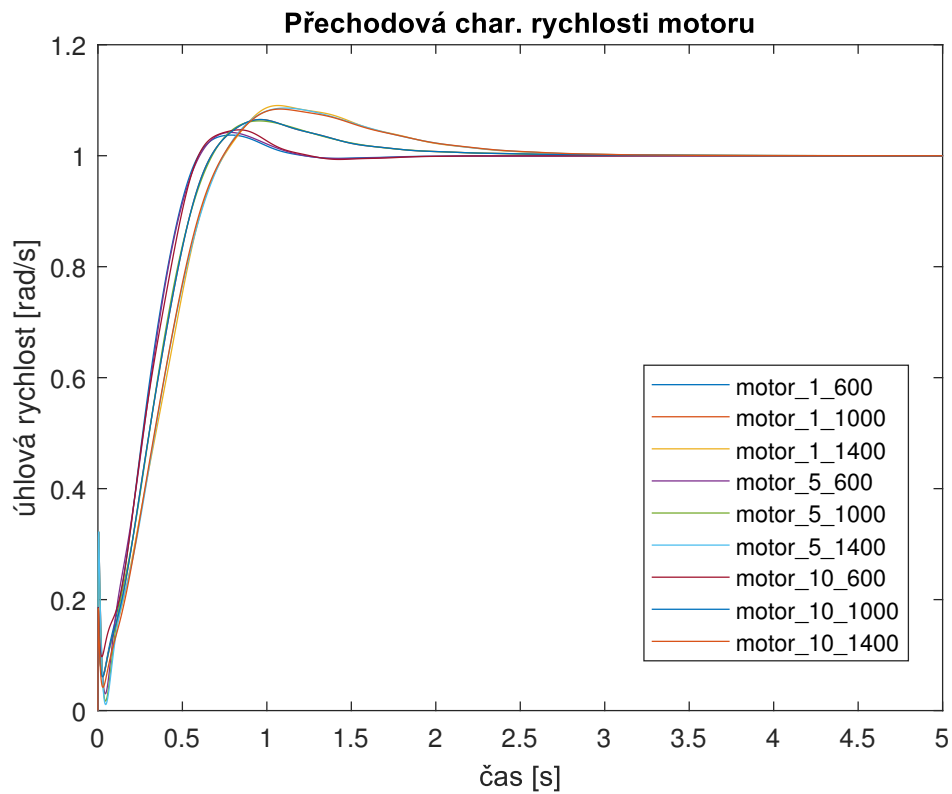
V následující kapitole se budeme věnovat generátoru trajektorie, který parametry regulace ještě vylepší, jelikož požadovaná hodnota regulátoru nebude generována skokově, nýbrž postupně po malých skocích (po diskretizované požadované křivce).



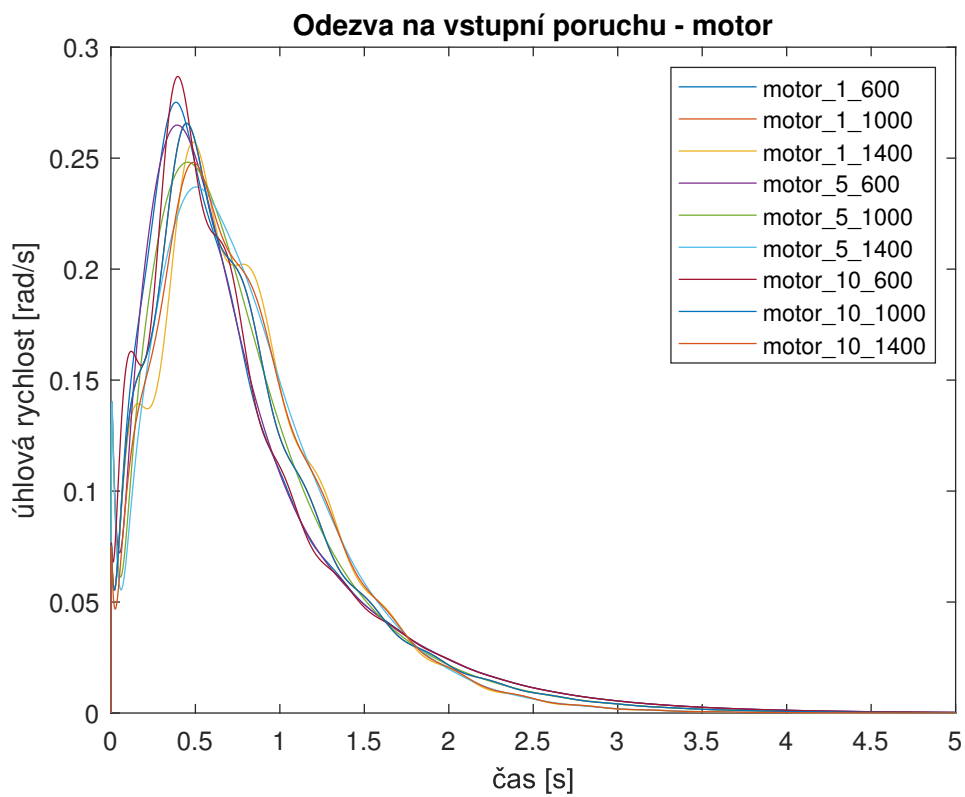
Obrázek 11: Přechodová charakteristika na straně kabiny, všech 9 systémů



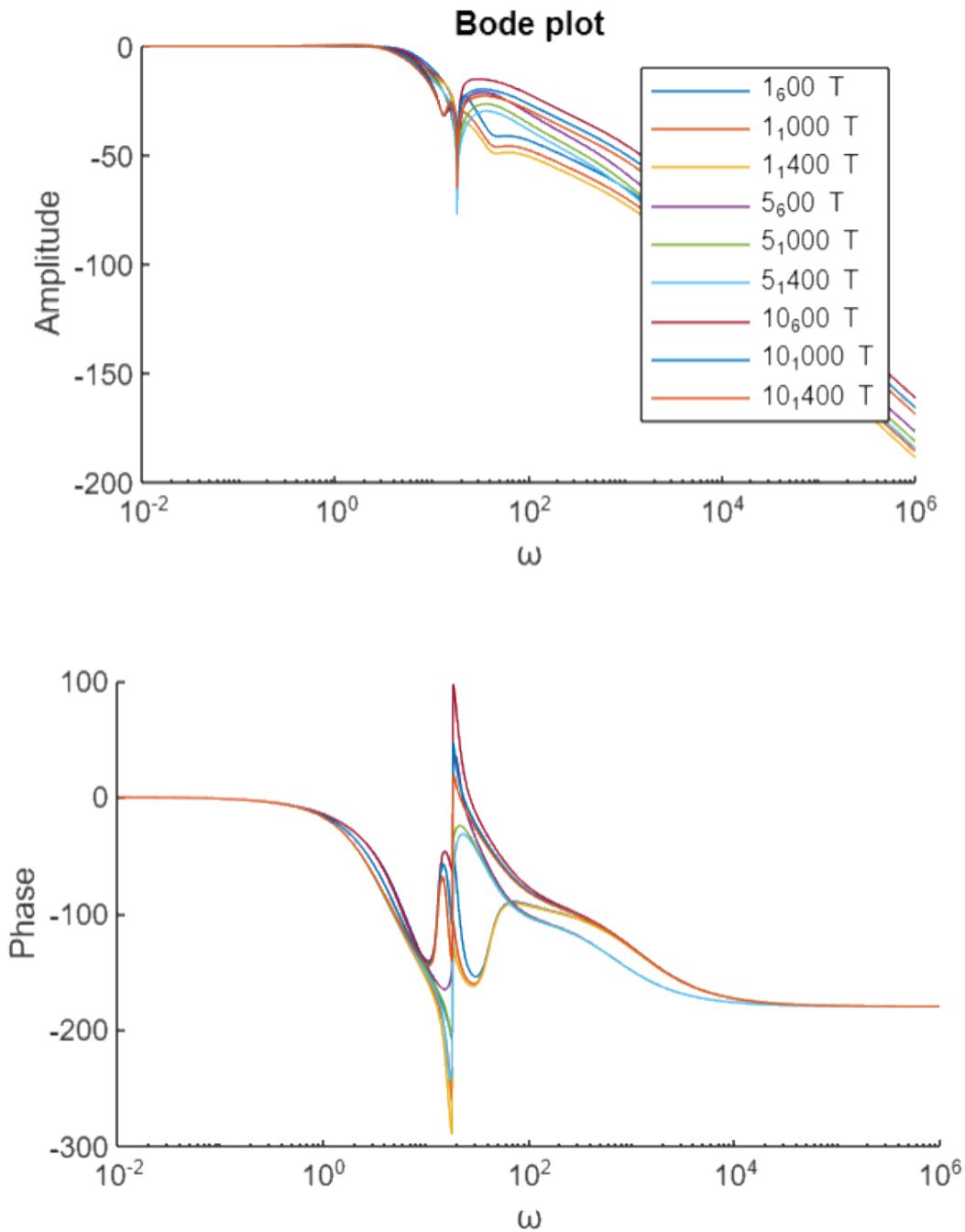
Obrázek 12: Odezva na vstupní poruchu na straně kabiny, všech 9 systémů



Obrázek 13: Přechodová charakteristika na straně motoru, všech 9 systémů



Obrázek 14: Odezva na vstupní poruchu na straně motoru, všech 9 systémů



Obrázek 15: Bodeho charakteristika na straně kabiny v aplikaci H_∞ , všech 9 systémů [2]

[2] Aplikace H_∞ bohužel neumožňuje export grafů, natož vektorově, proto přikládáme alespoň snímek aplikace, byť v nevalné kvalitě

2.7 Generátor trajektorie

Generátor trajektorie byl v projektu [1] zpracován kolegou Tvrzem na základě práce [7]. Nebudu jej zde proto rozvádět do zbytečných detailů.

Princip generátoru je ve své podstatě jednoduchý - předem vytvoří dráhu, po které se bude systém (výťah) pohybovat, a to ve formě křivek polohy, rychlosti, zrychlení a případně i jerku (3. derivace polohy). Činí tak na základě vstupních parametrů - počáteční a koncová poloha, maximální rychlost, maximální zrychlení a případně maximální jerk.

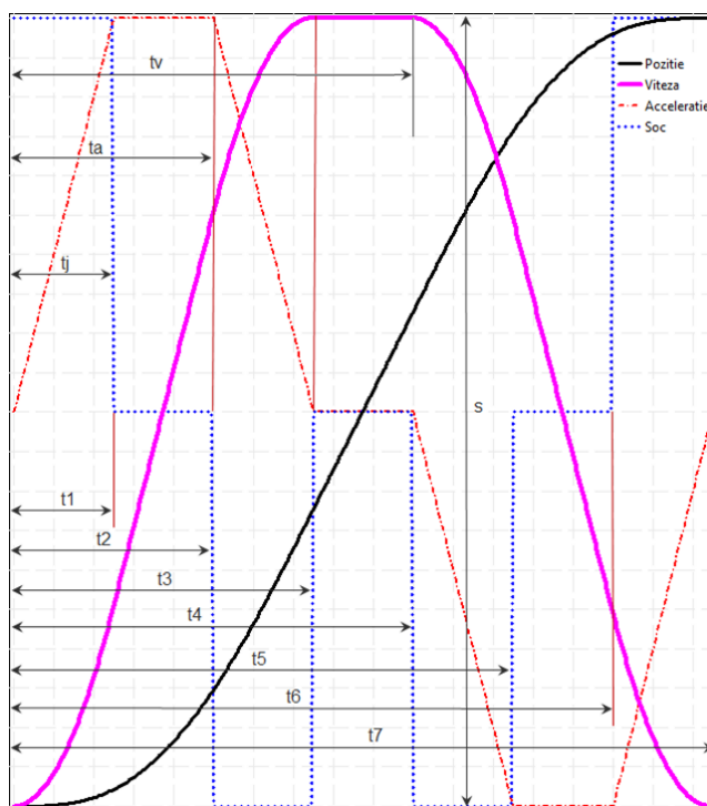
Regulátor pak nemusí reagovat na skokové změny ale na kvazispojité signály (stále se jedná o diskrétní posloupnost hodnot generovanou počítačem, oproti jednomu skoku požadované hodnoty se však jeví jako spojitá křivka).

Zmíněné křivky chceme generovat pokud možno co nejhladší, čímž se významně sníží generované oscilace. I díky tomu je dosaženo lepších parametrů regulace a tedy i komfortu uživatelů.

Jelikož my chceme generovat i trajektorii zrychlení (tz. nebude se měnit skokově, ale postupně díky integraci jerku), budeme mít 6 způsobů generování trajektorie [7, p. 15].

Vzhledem ke vstupním parametrům $(x_{20}, x_{21}, v_{max}, a_{max}, j_{max})$ se určí typ trajektorie a vypočítá se 7 přepínacích časů (mohou vzájemně splýnout, čímž interval mezi nimi zanikne), ve kterých se skokově mění jerk a ten se následně integruje na zrychlení, to dále na rychlost a následně na polohu.

$$j \xrightarrow{f} a \xrightarrow{f} v \xrightarrow{f} s \quad (30)$$



Obrázek 16: Znázornění přepínacích časů [7, p. 8]

Tímto způsobem tedy můžeme generovat hladké S-křivky trajektorie.

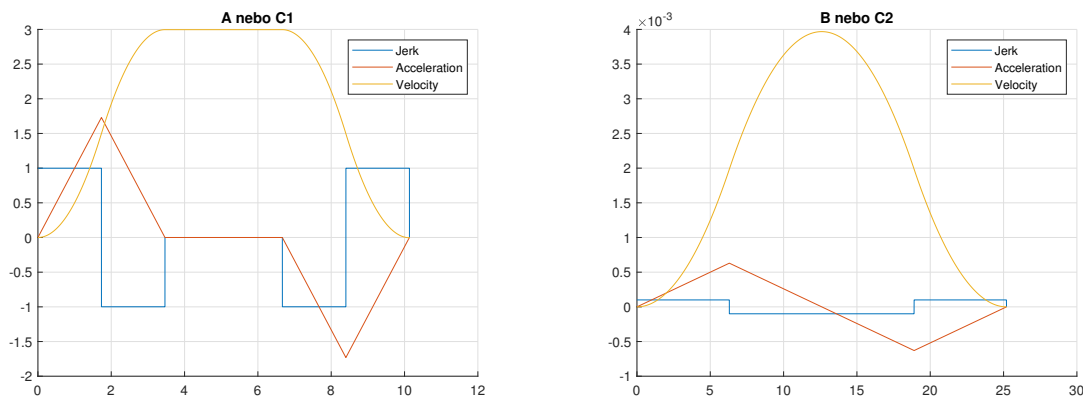
Následně musíme generované křivky (časy přepnutí jerku) diskretizovat tak, aby souhlasily s periodou vzorkování regulátoru (v opačném případě by došlo k přepnutí až v moment vzorku a výtah by nedojel na požadovanou polohu přesně). To učiníme tak, že pokud není vygenerovaný čas přepnutí dělitelný periodou vzorkování, zaokrouhlí se právě na periodu vzorkování, a to buď nahoru, nebo dolů. O tom, na jakou stranu zaokrouhlení proběhne, rozhodne fakt, že nikdy nesmíme překročit původní omezení.

Toto se pokusíme demonstrovat na jednoduchém příkladu:

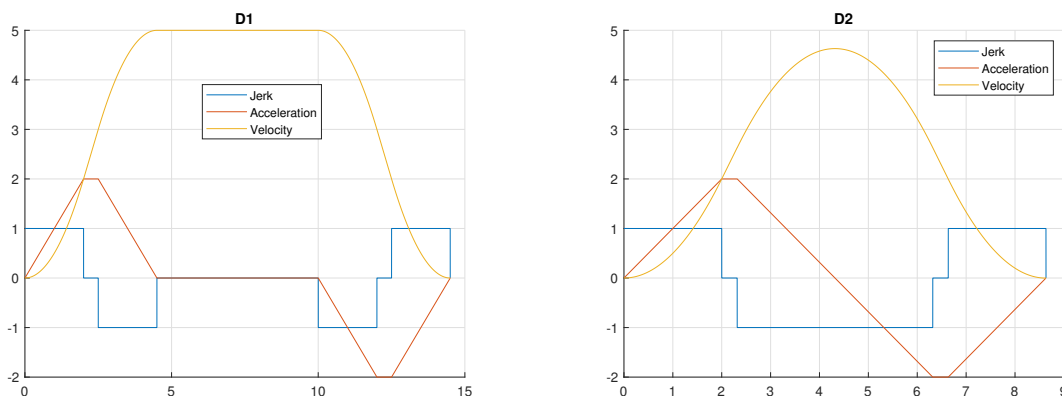
$$\left. \begin{array}{l} t = s/v_max_original; \\ t = t - \text{mod}(t, \text{vzorek}) + \text{vzorek}; \\ v_max_new = s/t; \end{array} \right\} \Rightarrow \begin{array}{l} v_max_new < v_max_original \\ t \in \text{perioda vzorkování} \end{array} \quad (31)$$

V praxi to znamená, že výtah přepne jerk později než byl vygenerovaný přepínací čas, a tudíž aby dojel do stejné pozice, musí jet nižší maximální rychlostí.

Jak již bylo dříve zmíněno, máme 6 možností generování trajektorie. Ve výsledku budou ale pouze 4 profily pohybu^[3]. Například na [obrázku 17a](#) dojde k omezení (saturaci) rychlosti - výtah se po nějakou dobu pohybuje s konstantní rychlostí. K tomu dojde, vzhledem k vnitřní matematické implementaci, ze dvou důvodů (proto značení A nebo C1), které zde však nedokážu dostatečně stručně popsat (detailně v [7, p. 12-14]). Vykrešleno včetně zohlednění diskretizace.



Obrázek 17: Možnosti A nebo C1 a B nebo C2



Obrázek 18: Možnosti D1 a D2

^[3]Označení (abecední) na obrázcích 18 a 17 odpovídá práci [7, p. 15]: A/C1-došlo k omezení rychlosti, B/C2-nedošlo k omezení rychlosti ani zrychlení, D1-došlo k omezení rychlosti i zrychlení, D2-došlo k omezení zrychlení ale ne rychlosti

2.8 Simulace robustní regulace na nelineárním modelu s generátorem trajektorie

Nyní potřebujeme navržené metody poskládat do jednoho celku. Na [obrázku 19](#) je vidět celkové schéma, kde je nelineární model (popsaný pohybovými rovnicemi, viz vztahy 6 až 8), generátor trajektorie (ve formě Matlab function) a výše popsany PI regulátor. Jelikož výtah řídíme pomocí motoru, jsou vstupní parametry GT omezení na straně motoru (tedy maximální rychlost otáčení motoru, maximální zrychlení atd. - v realitě budeme znát spíše omezení motoru než kabiny). Počáteční a koncová poloha motoru (ϕ_0, ϕ_1) je přepočtena z požadovaných počátečních a koncových poloh kabiny (x_{20}, x_{21}).

V celé této části provádíme simulace s následujícími parametry:

$$\text{Hmotnost kabiny: } m_2 = 1000 \text{ [kg]}$$

$$\text{Počáteční poloha kabiny: } x_{20} = 6 \text{ [m]}$$

(z té již vyplývají ostatní počáteční podmínky)

$$\text{Maximální rychlost motoru: } 100 \text{ [rad} \cdot \text{s}^{-1}\text{]}$$

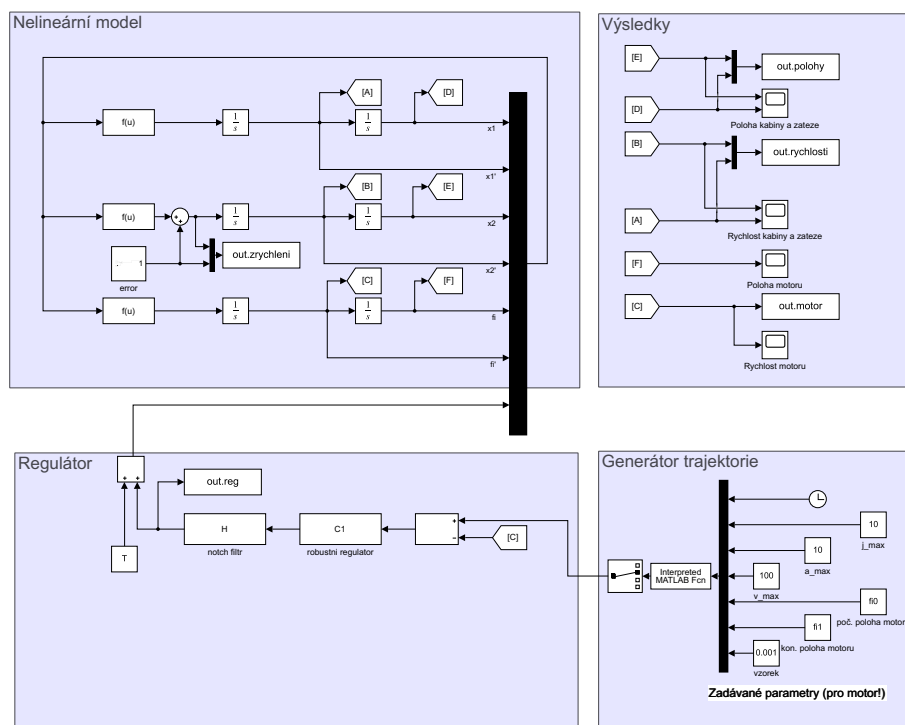
$$\text{Maximální zrychlení motoru: } 10 \text{ [rad} \cdot \text{s}^{-2}\text{]}$$

$$\text{Maximální jerk motoru: } 10 \text{ [rad} \cdot \text{s}^{-3}\text{]}$$

$$\text{Cílová poloha kabiny: } x_{21} = 25 \text{ [m]}$$

$$\text{Vzorkování: } 0.001 \text{ [s]}$$

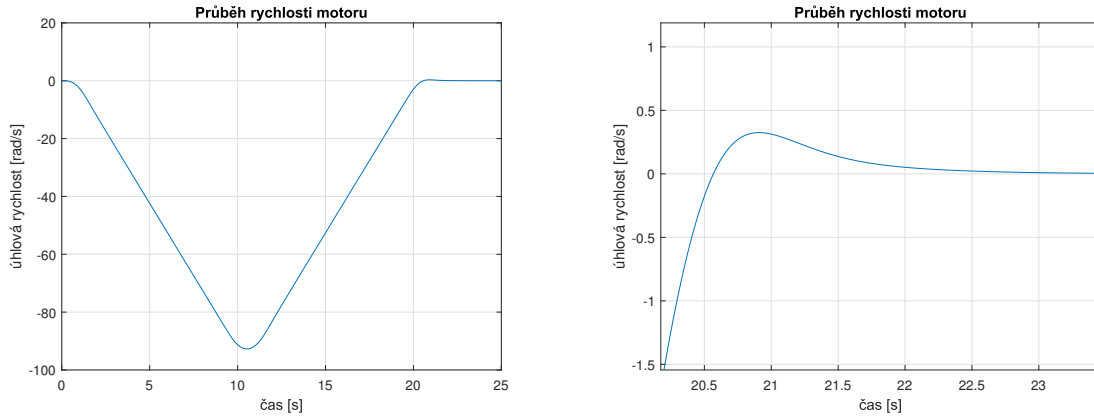
(32)



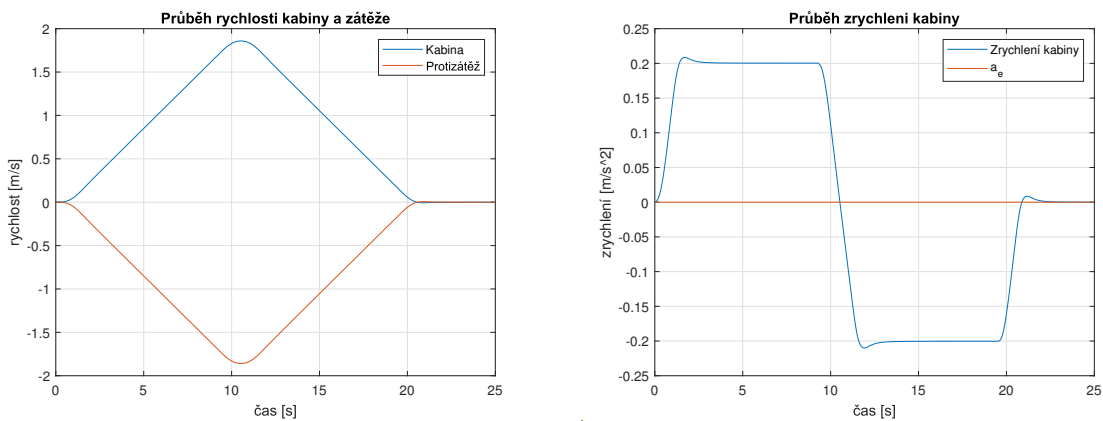
Obrázek 19: Schéma modelu s generátorem trajektorie v Simulinku

Výsledná regulace rychlosti motoru viz [obrázek 20](#). Překmit je zde díky GT výrazně snížen.

Rychlost kabiny a protizávaží je vidět na [obrázku 21a](#). Zrychlení kabiny výtahu je pak vidět na [obrázku 21b](#) společně s chybou a_e působící na kabinu (zde nulová, využíváme ji dále v [sekci 8](#)).



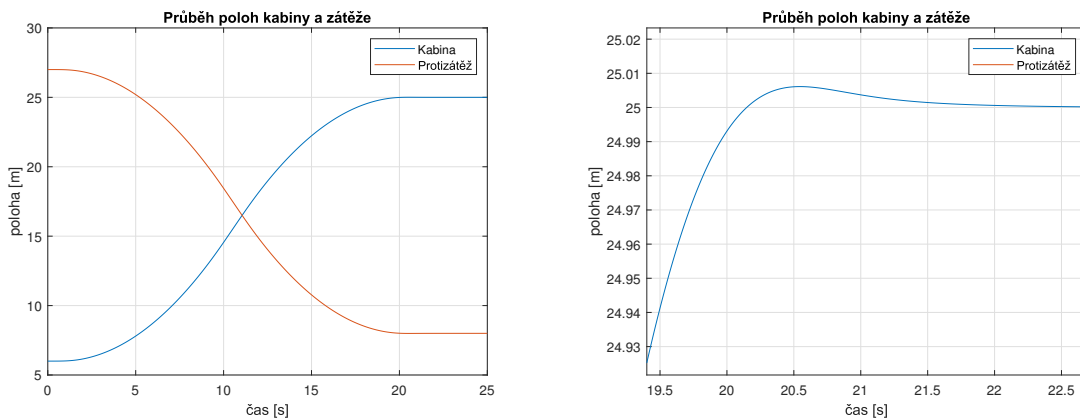
Obrázek 20: Robustní regulace - rychlost motoru a detail (vpravo)



(a) Rychlost kabiny a protizávaží

(b) Zrychlení kabiny

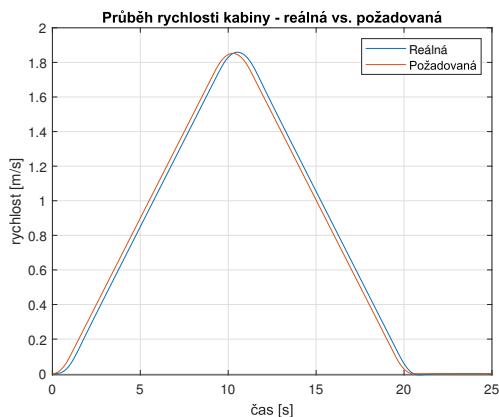
Obrázek 21: Robustní regulace - rychlosti a zrychlení



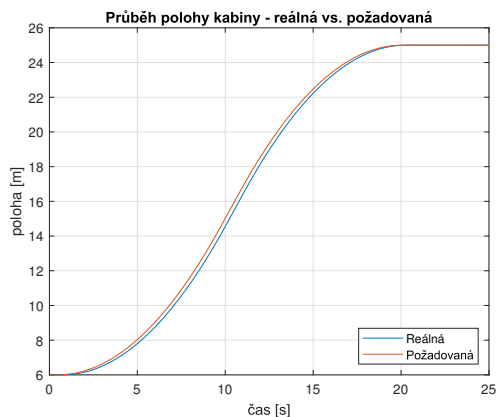
Obrázek 22: Robustní regulace - polohy kabiny a zátěže + detail (vpravo)

Překmit u polohy (obrázek 22) činí pouze 5 mm. V praxi by bylo vhodné ještě zavést kaskádní regulaci přes polohu, jelikož v případě odlišného tření oproti uvažovanému, nebo v případě „poskočení“ lidí ve výtahu, se změní průběh rychlosti a tudíž i výsledná poloha. Druhou možností je neregulovat přesně na cílovou polohu s nulovou koncovou rychlostí, ale poblíž ní s rychlostí nízkou, přičemž výtah se „došoupe“ do požadované polohy, kde narazí na mechanické dojezdové spínače. Tím se rovněž provede autokalibrace.

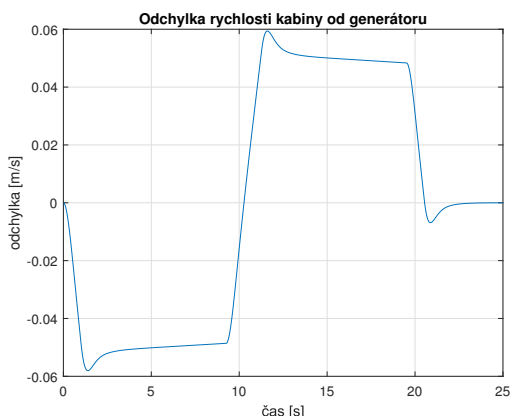
Pro porovnání s následujícími metodami řízení uvedeme ještě odchylky skutečné rychlosti a polohy od té požadované z generátoru trajektorie - obrázek 23.



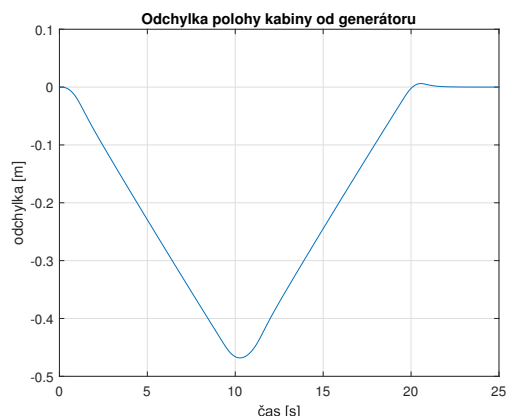
(a) Rychlost kabiny pož. vs. reálná



(b) Poloha kabiny pož. vs. reálná



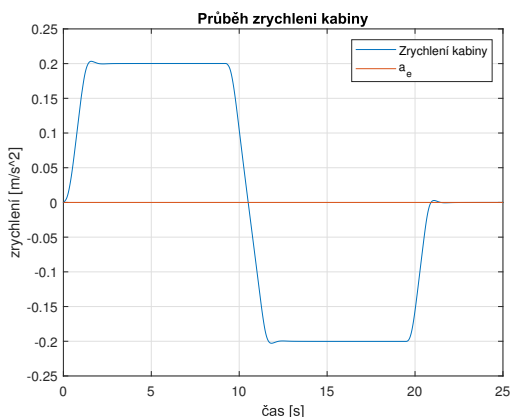
(c) Odchylka rychlosti kabiny od pož.



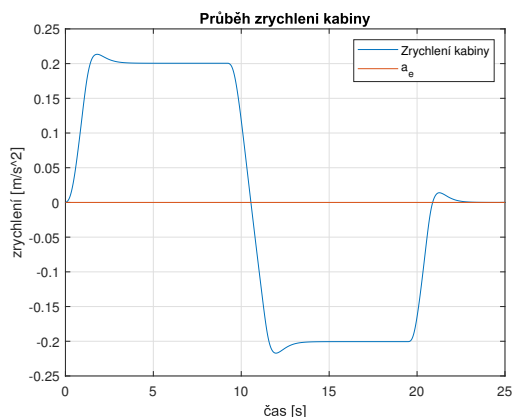
(d) Odchylka polohy kabiny od pož.

Obrázek 23: Robustní regulace - odchylky od generátoru

Na obrázku 24 je pak vidět rozdíl ve zrychlení kabiny mezi vahami 600kg a 1400kg při této robustní regulaci.



(a) Zrychlení kabiny - $m_2 = 600kg$



(b) Zrychlení kabiny - $m_2 = 1400kg$

Obrázek 24: Robustní regulace - zrychlení kabiny pro $m_2 = 600kg$ a $m_2 = 1400kg$

2.9 Závěr projektu

Podářilo se nám získat vhodný robustní regulátor, doplněný notch filtrem a generátorem trajektorie, který zvládne uřídít nejen linearizovaný model, ale i nelineární simulaci výtahu, a to pro různá zatížení a pro různé scénáře pohybu mezi rozličnými patry.

3 Gain-scheduling

Regulátor získaný v předchozí kapitole nyní ještě vylepšíme metodou gain-scheduling (GS).

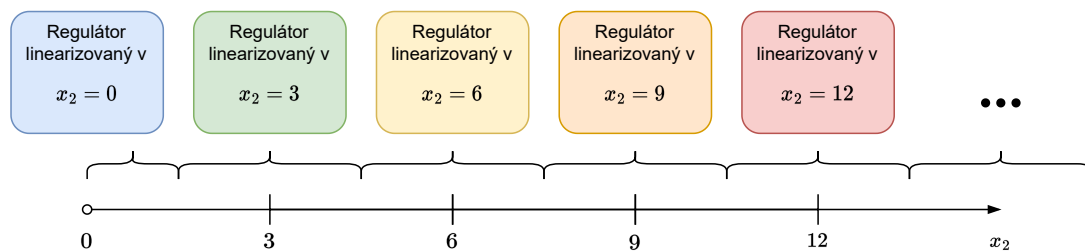
Tato metoda využívá sledovatelné proměnné vlastnosti systému k úpravě parametrů regulátoru v průběhu regulace. Jelikož v našem případě se mění parametry dva (zatížení a výška), přičemž sledovatelný je pouze jeden (výška), musíme i přesto navrhnout regulátory dostatečně robustní. Můžeme však již eliminovat jednu neurčitost, tudíž by se měly alespoň částečně vylepšit vlastnosti regulace, jmenovitě se díky zmenšení neurčitosti může regulátor zagresivnit, čili bude ve svých zásazích méně konzervativní, měla by se zvětšit šířka pásma i zlepšit kompenzace náhodných poruch.

Jak píší W. Sienel a T. Bunte [8, p. 554], konvenční přístup spočívá v návrhu regulátorů pro konečný počet RB, přičemž zesílení regulátoru je pak mezi těmito body interpolováno.

Budeme tedy postupovat takto: Máme 10 pater, proto pro každé patro navrhne pro systém H_∞ jeden robustní regulátor pro 3 dosud uvažovaná zatížení (600, 1000 a 1400 kg) podle ITAE kritéria zátěže (kabiny) pro střední zatížení (1000 kg).

Nutno dodat, že, na rozdíl od předchozí kapitoly, tyto regulátory navrhujeme bez přítomnosti notch filtru! Díky tomu se dále vylepší regulace (regulátor 'uvidí' i poruchy na frekvencích, které by jinak byly potlačeny filtrem).

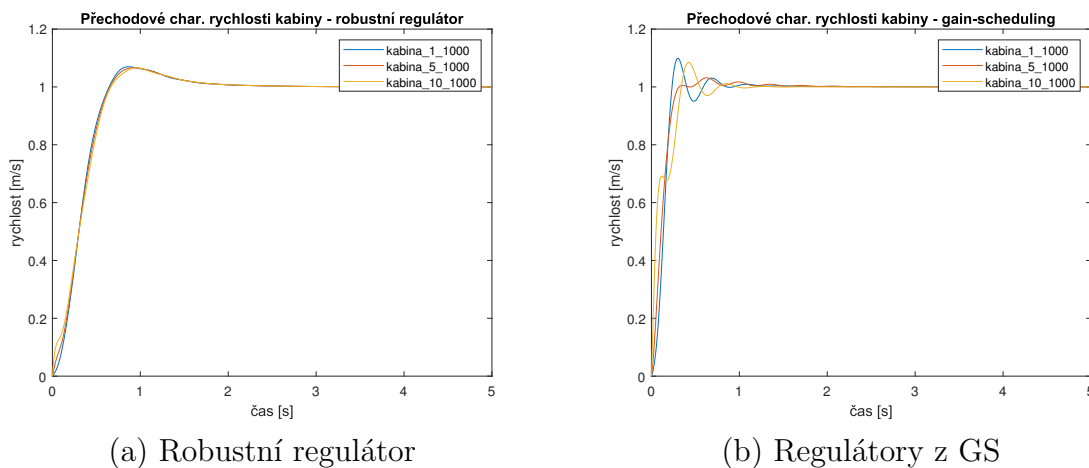
Získáme tedy 10 'lehce robustních' regulátorů, místo jednoho 'velmi robustního'. Ty by tedy měly pro své příslušné patro fungovat lépe, než robustní regulátor, ale v jiných patrech zase hůře, proto se mezi nimi bude přepínat.



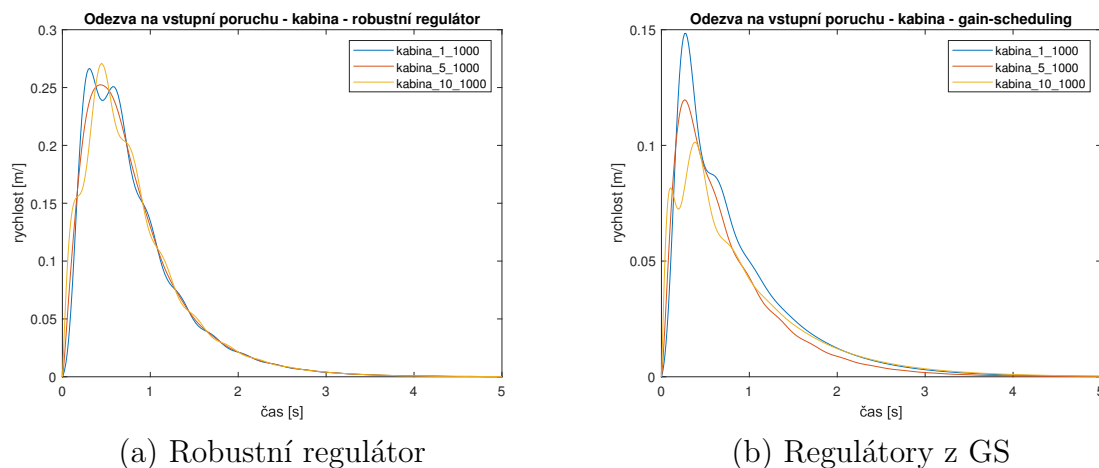
Obrázek 25: Princip fungování gain-schedulingu

To můžeme ověřit z příslušných charakteristik (pouze pro kabinu). Charakteristiky jsme vygenerovali pro 1., 5. a 10. patro, vždy pouze pro zatížení 1000kg (kvůli přehlednosti, navíc jsou na tuto váhu navrženy regulátory z GS). Na [obrázku 26a](#) můžeme vidět přechodové charakteristiky systémů řízených jedním robustním regulátorem, pro porovnání pak výrazně agresivnější a rychlejší odezva regulátorů z GS (bez notch filtru) na [obrázku 26b](#). V tomto případě reguluje v každém patře příslušný regulátor, tedy přenos *kabina_1_1000* je řízen regulátorem získaným pro 1. patro, přenos *kabina_5_1000* je řízen regulátorem získaným pro 5. patro atd.

Obdobně pak odezva na jednotkovou vstupní poruchu na obrázcích [26a](#) a [26b](#).



Obrázek 26: Porovnání přechodové charakteristiky mezi robustní regulací a GS



Obrázek 27: Porovnání odezvy na vstupní poruchu mezi robustní regulací a GS

Charakteristiky jsou samozřejmě tvořeny spojením regulátoru a příslušného linearizovaného přenosu na kabinu v daném patře, aplikace na nelineární systém může mít odezvu lehce odlišnou.

Následují frekvenční charakteristiky. Ty je nutno vytvářet pro přenos tvořený regulátorem a přenosem na rychlost motoru, protože právě přes motor je uzavřena zpětná vazba (nikoli přes kabinu).

Bodeho charakteristika (také zvaná logaritmická frekvenční) využívá znalosti přenosu otevřené smyčky ke zjištění parametrů stability smyčky uzavřené. Tato vazba ale musí být uzavřena přes stejný přenos, pro který vytváříme Bodeho charakteristiku, proto zde nelze využít přenos na rychlost kabiny (nekolokovaná zpětná vazba - z anglického *non-located feedback*), ale pouze přenos na rychlost motoru (kolokovaná zpětná vazba).

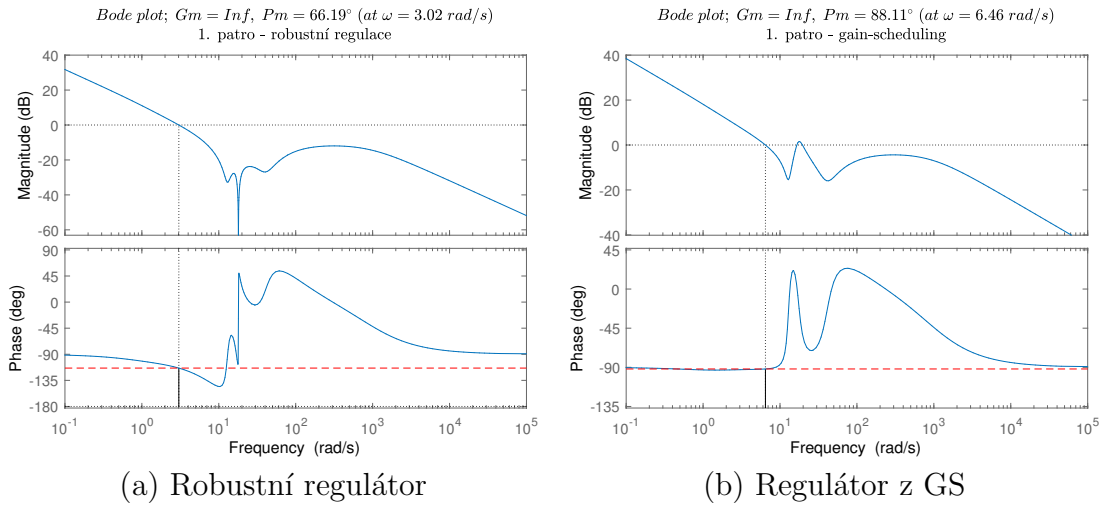
Rovněž Nyquistova křivka (v angličtině Nyquist plot) slouží ke zjištění parametrů uzavřené křivky na základě otevřené smyčky. Bude tedy též vytvořena pro spojení regulátoru a přenosu na motor.

V případě robustní regulace je do přenosů ještě zaveden notch filtr. U GS nikoli.

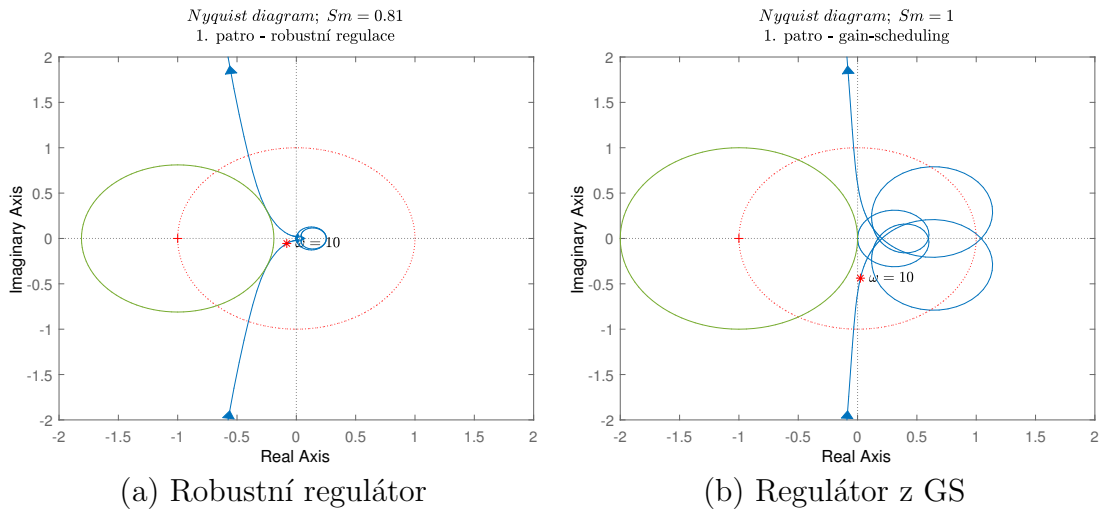
Jak jsme očekávali, regulátory navržené pro GS pro konkrétní patro dosahují významně lepších indexů stability (G_m , P_m i S_m - obrázky 28, 29, 31, 32, 34 a 35), jakož i vyšších hodnot šířky pásma ω_B (obrázky 30, 33 a 36), než robustní regulátor navržený pro všechna patra zároveň.

Tudíž v případě, že se nám povede přepínat regulátory hladce, dosáhneme významně lepší kvality řízení v celé délce.

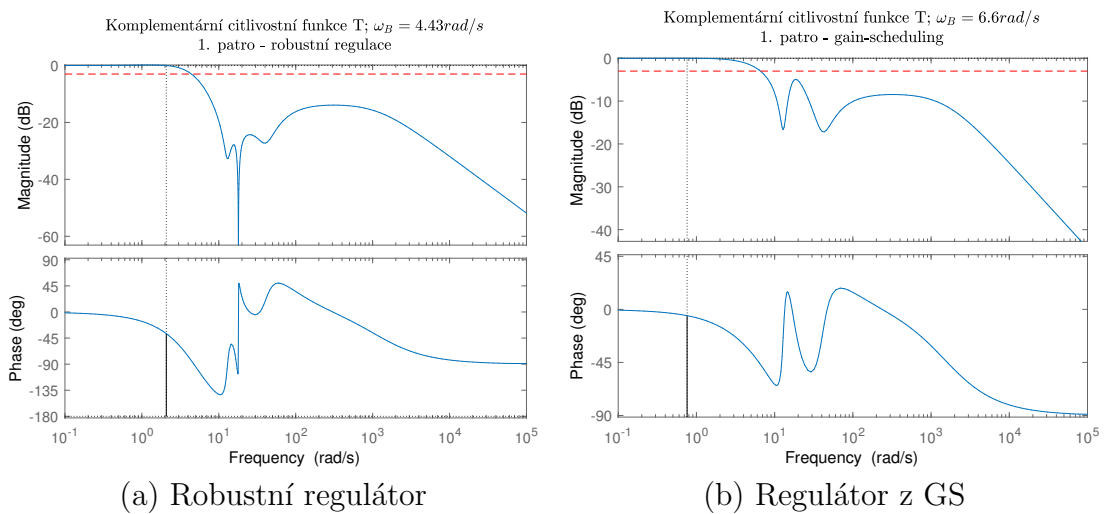
Charakteristiky pro 1. patro ($m_2 = 1000kg$):



Obrázek 28: Bodeho charakteristiky pro 1. patro

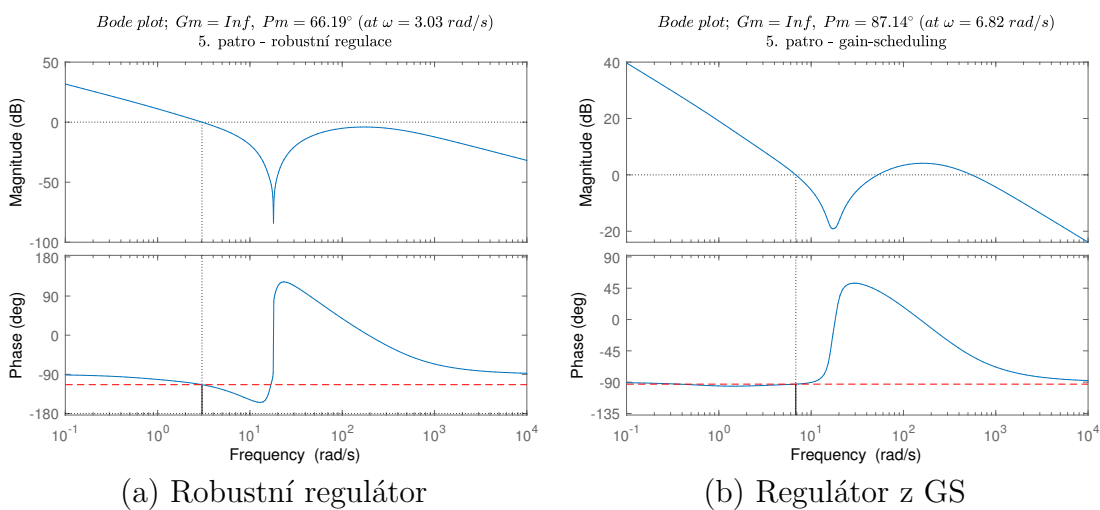


Obrázek 29: Nyquistova křivka pro 1. patro

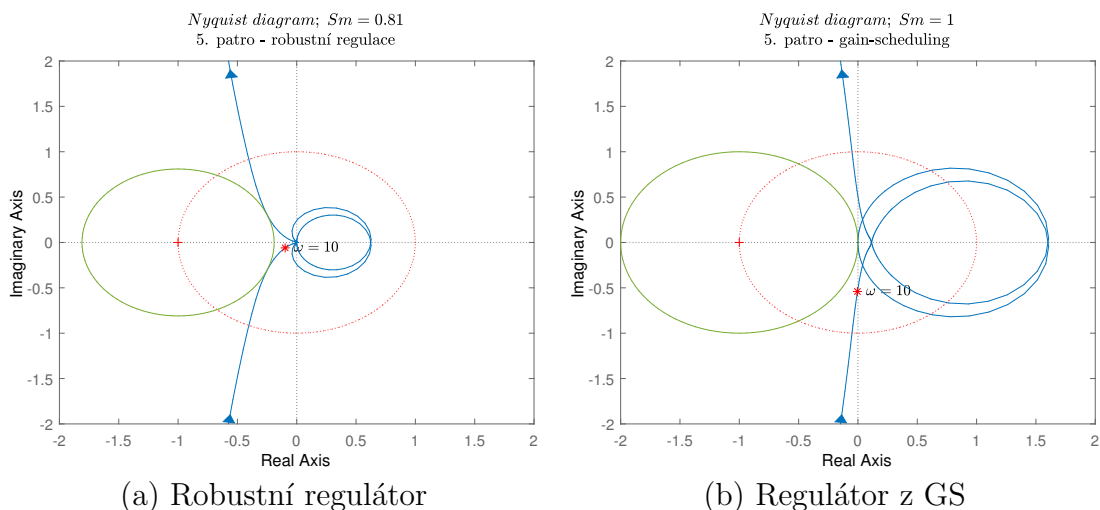


Obrázek 30: Komplementární citlivostní funkce pro 1. patro

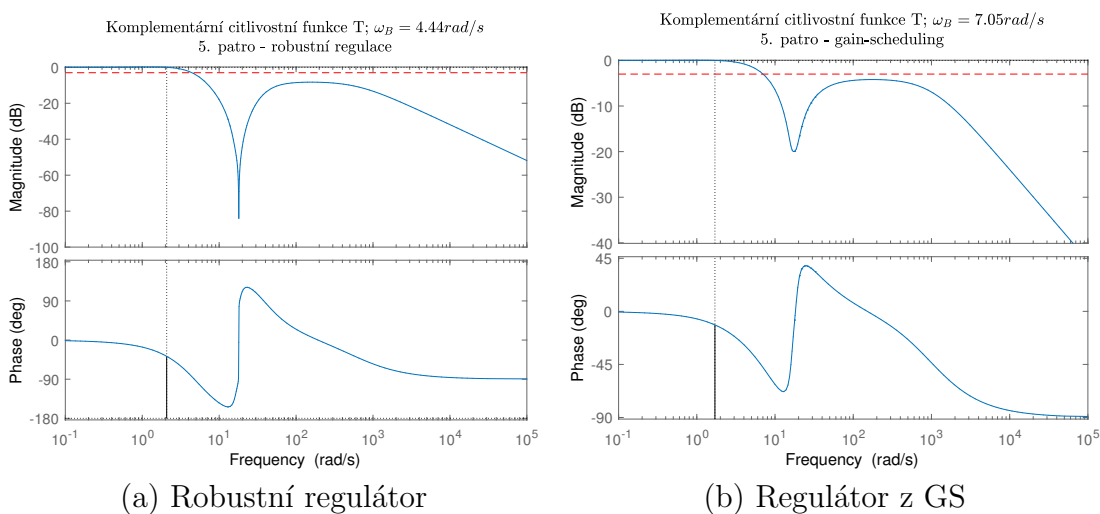
Charakteristiky pro 5. patro ($m_2 = 1000kg$):



Obrázek 31: Bodeho charakteristiky pro 5. patro

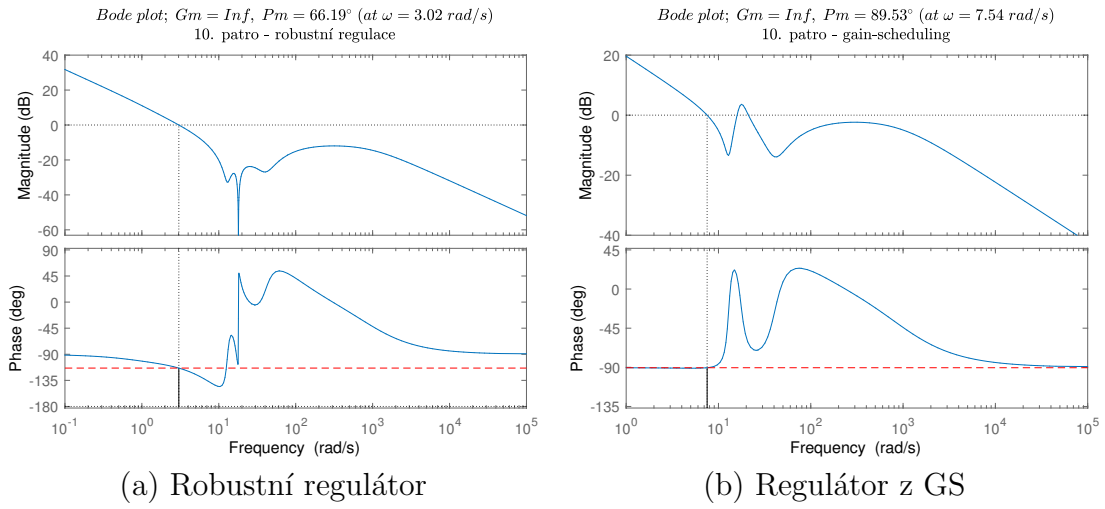


Obrázek 32: Nyquistova křivka pro 5. patro

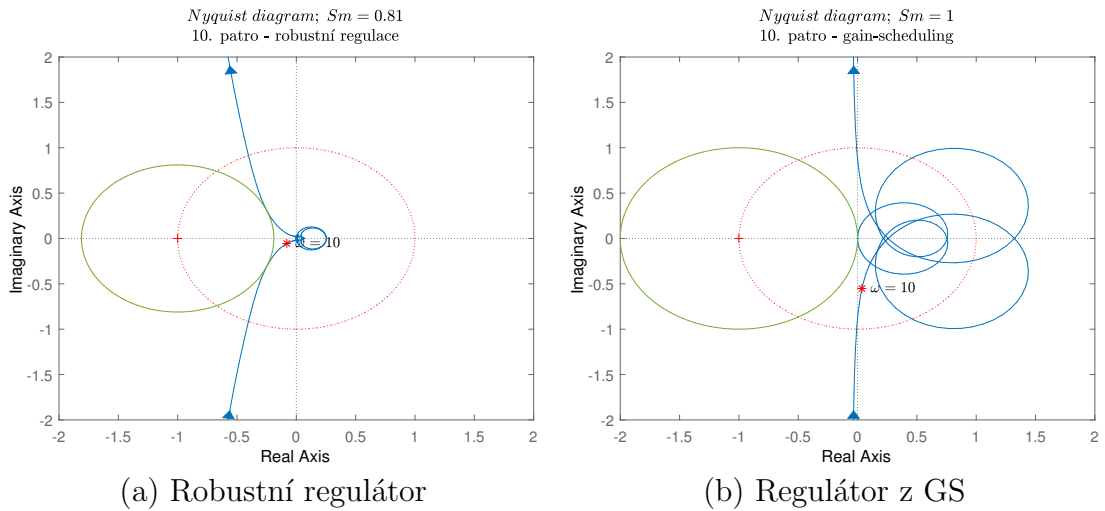


Obrázek 33: Komplementární citlivostní funkce pro 5. patro

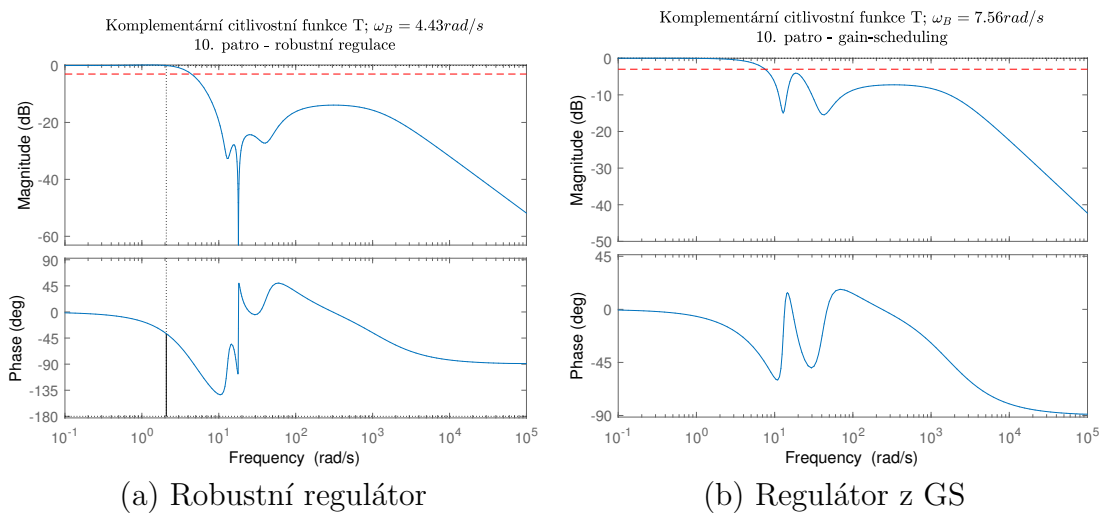
Charakteristiky pro 10. patro ($m_2 = 1000kg$):



Obrázek 34: Bodeho charakteristiky pro 10. patro



Obrázek 35: Nyquistova křivka pro 10. patro

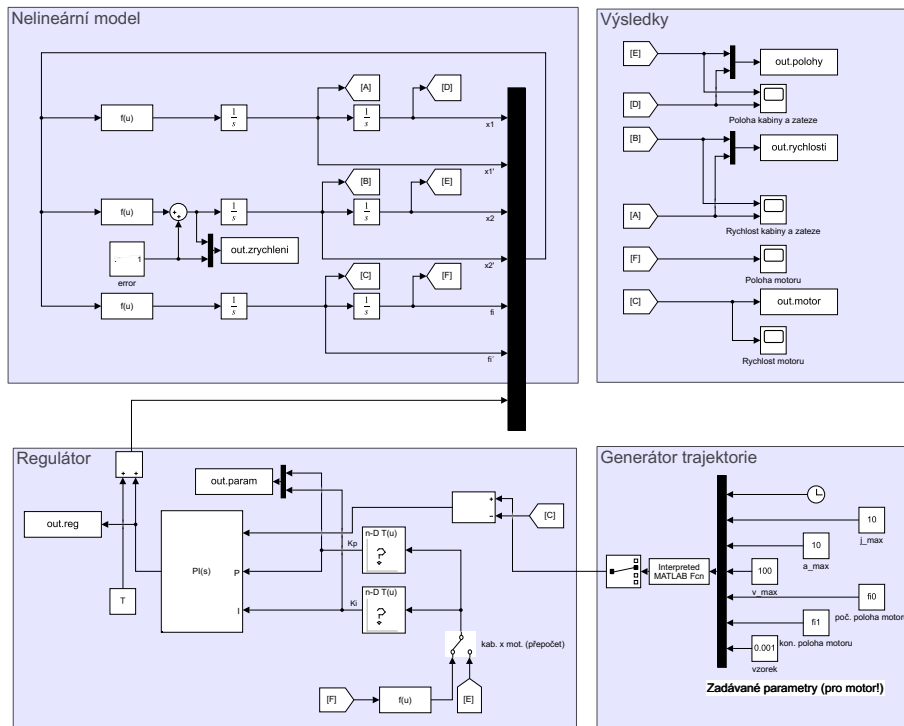


Obrázek 36: Komplementární citlivostní funkce pro 10. patro

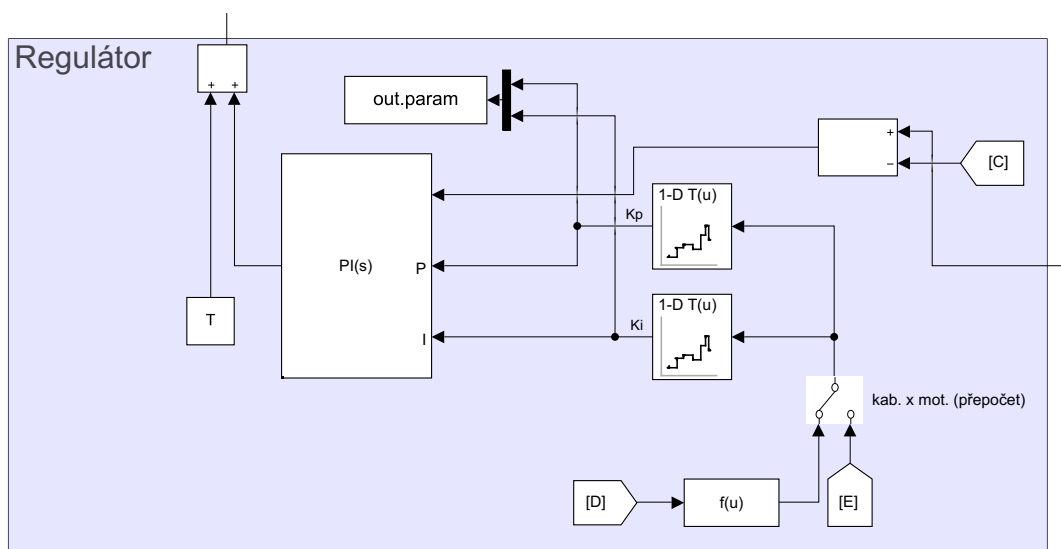
V Simulinku gain-scheduling implementujeme pomocí bloků Lookup Table a PID Controller, viz schémata na obrázcích 37 a 38.

Princip je tedy ten, že v závislosti na poloze kabiny x_2 se v tabulkách vyhledají příslušná zesílení K_i, K_p regulátoru, který byl navržen pro linearizaci ve výšce nejbližší momentální x_2 . Parametry PI regulátoru se tedy mění průběžně.

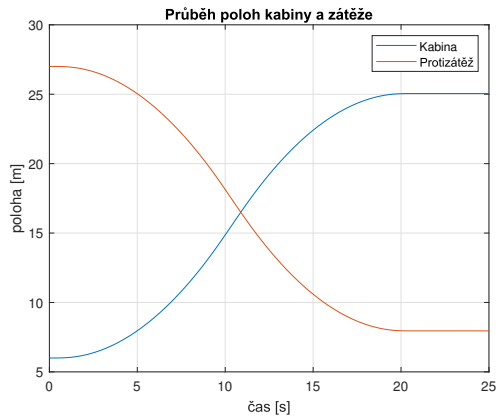
Přepínač u regulátoru (obrázek 38) slouží k nastavení sledování rychlosti kabiny přímo nebo přes přepočít od motoru (výsledky jsou totožné).



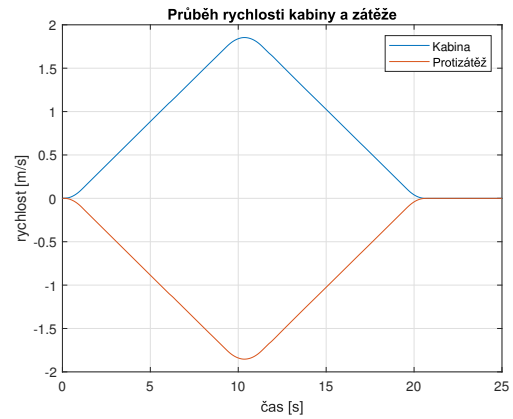
Obrázek 37: Simulační model pro gain-scheduling



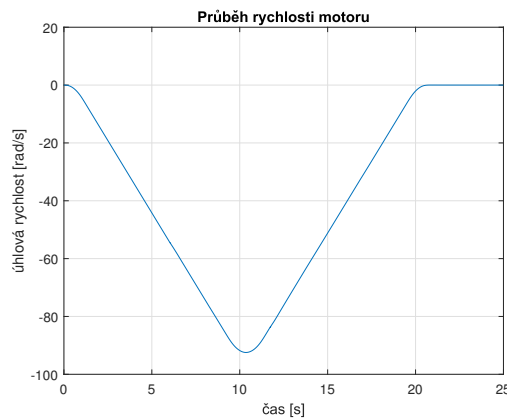
Obrázek 38: Simulační model pro gain-scheduling - detail regulátoru



(a) Polohy kabiny a zátěže



(b) Rychlosti kabiny a zátěže



(c) Rychlost motoru

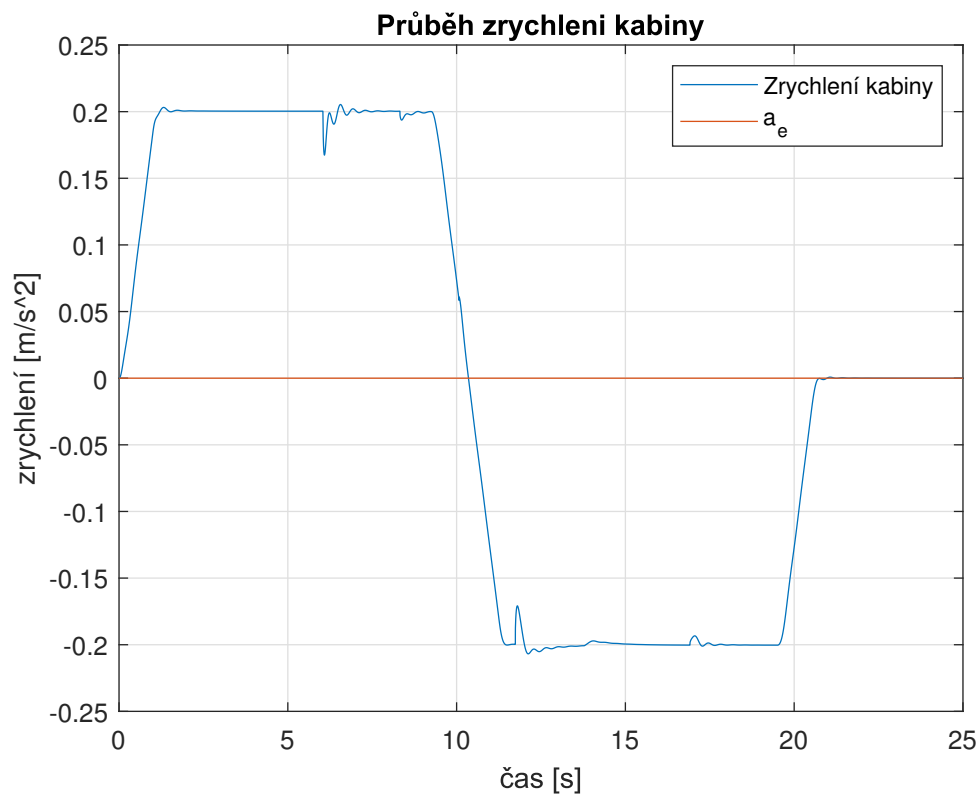
Obrázek 39: Gain-scheduling

U průběhu poloh (obrázek 39a) a rychlostí (obrázek 39b) kabiny a protizávaží se tato regulace jeví jako dobrá. Problém nastává u zrychlení, kde dochází k rázům, viz obrázek 40.

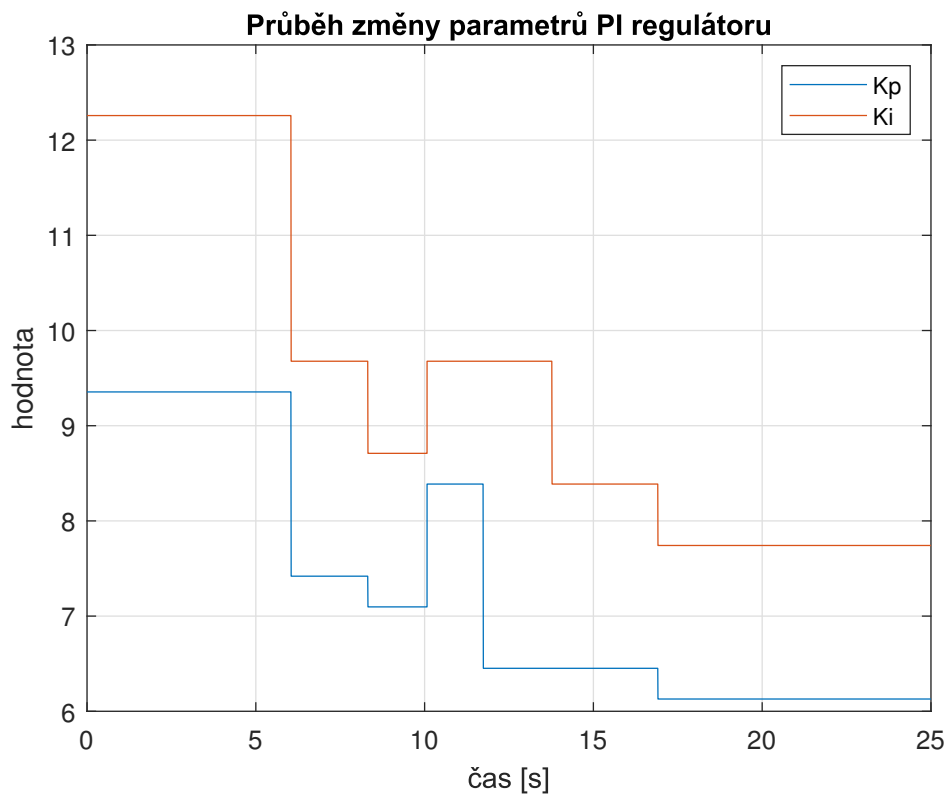
Tato implementace má totiž jeden zásadní problém - při změnách parametrů K_p , K_i dochází k nespojitostem ve výstupu regulátoru, viz obrázky 42 a 43.

To je důsledkem přepnutí regulátoru (jak je možno vidět z obrázku 41), což matematicky odpovídá nekomutativnosti při násobení integrálu časově proměnnou funkcí.

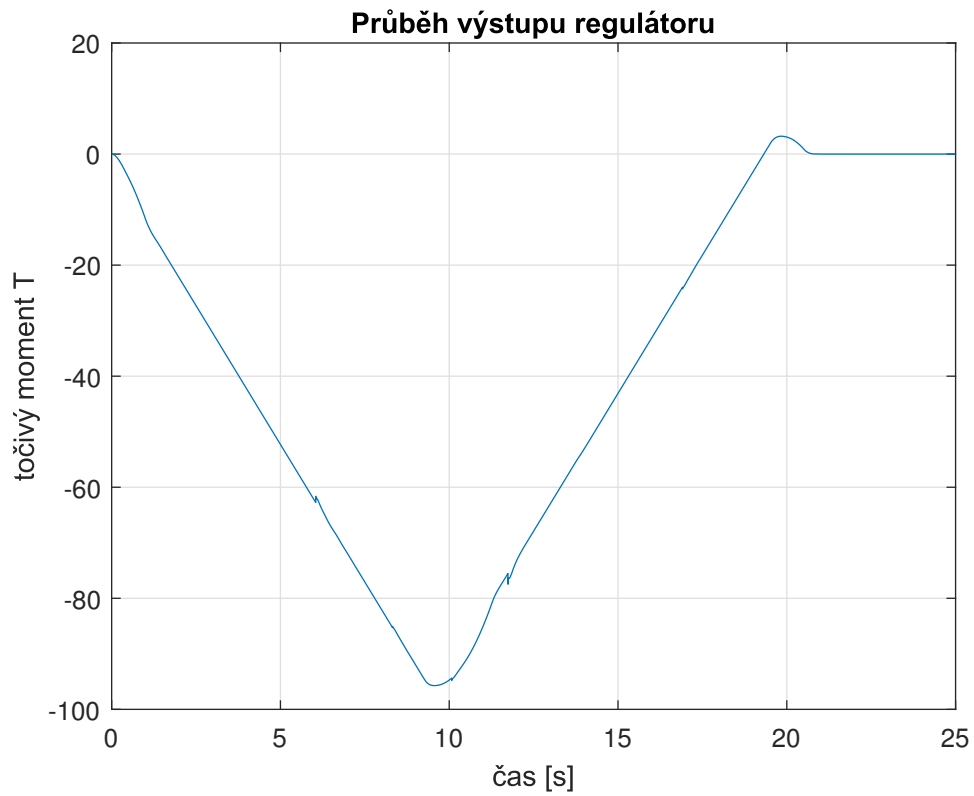
$$I = K_i(t) \int_0^t e(\tau) d\tau \neq \int_0^t K_i(t) e(\tau) d\tau \quad (33)$$



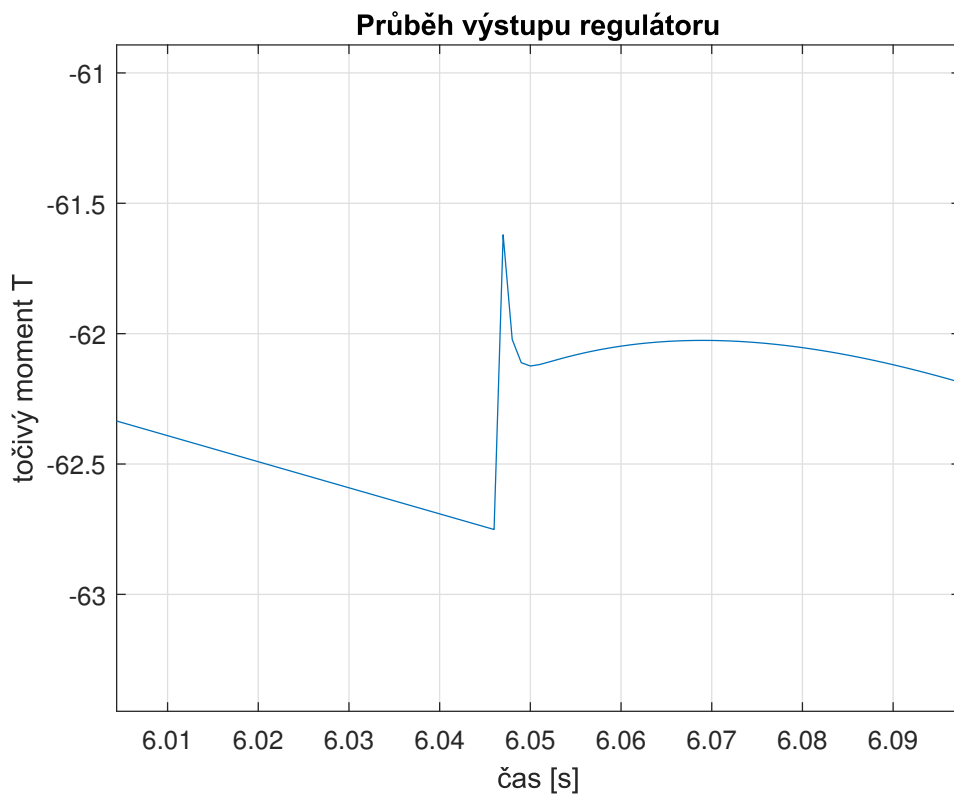
Obrázek 40: Gain-scheduling - zrychlení kabiny



Obrázek 41: Gain-scheduling - změna parametrů regulátoru

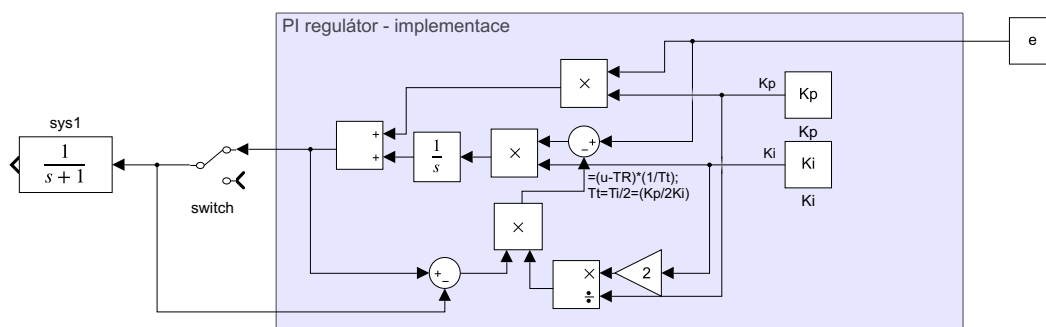


Obrázek 42: Gain-scheduling - nespojitý výstup regulátoru



Obrázek 43: Gain-scheduling - nespojitý výstup regulátoru, detail

V praxi se toto ošetřuje vysledováním výstupu regulátoru. Problém s naší implementací je ale ten, že my nepřepínáme mezi jednotlivými regulátory, ale měníme parametry jednoho regulátoru. Pokud bychom tedy zavedli vysledování výstupu, nebude to mít vůbec žádný vliv, viz. schéma na [obrázku 44](#), protože sledovaný výstupní signál se okamžitě odečte s výstupem, tedy od odchylky před integrátorem se odečte 0, čili jako kdyby vysledování vůbec neexistovalo. Je to tím, že u klasické implementace GS by bylo 10 jednotlivých PI regulátorů ovládaných přes switch (selectoring, na schématu vyznačeno slovem switch), tudíž regulátor, na který se zrovna přepnulo, by se nastavil podle starého (rozdíl signálů by nebyl nulový). Zde však měníme parametry, což je změna, která nelze vysledovat jinak, než že by se poslední hodnota výstupu *někam* uložila, následně by se vnitřní integrátor vynuloval a poté se *nějak* naintegroval zpět tak, aby byl celkový výstup stejný jako před přepnutím (složka K_p je též jiná).



Obrázek 44: Problém s vysledováním u této implementace GS

Nyní máme tedy tři možnosti, jak tento problém vyřešit:

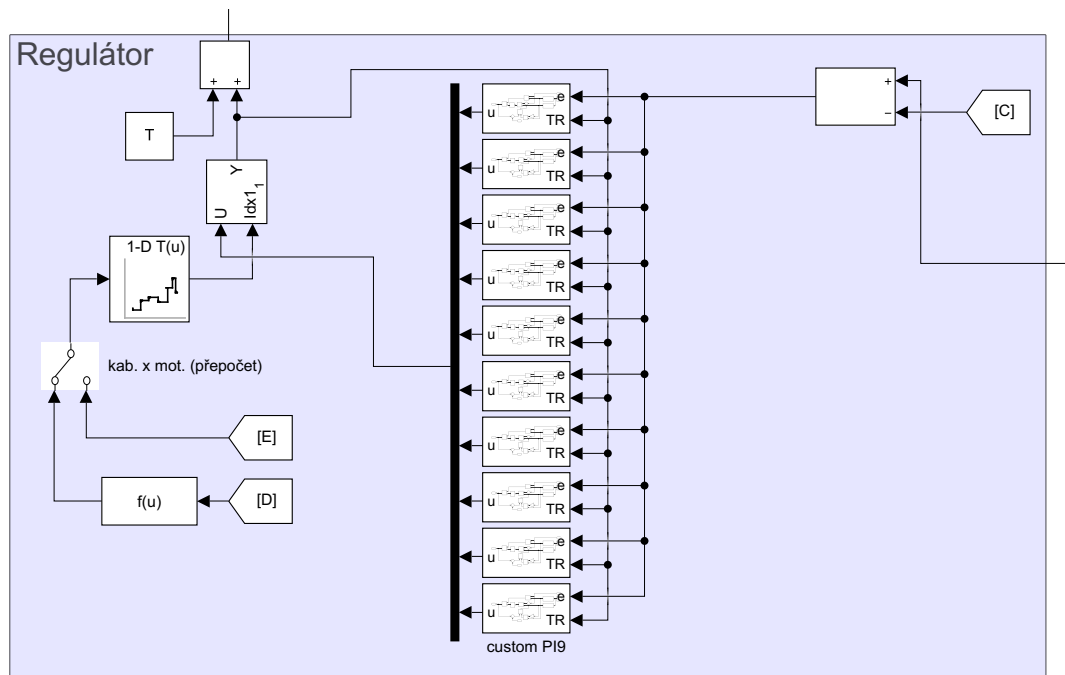
1. Naimplementovat 10 PI regulátorů, mezi kterými budeme přepínat switchem (selectoring), čímž zprovozníme vysledování.
2. Neměnit hodnoty K_p a K_i skokově, ale proložit je spojitou hladkou křivkou.
3. Využít toho, že regulátor je potřebný pouze k rozjezdu a brzdění, protože v průběhu jízdy jsou jeho zásahy minimální, a přepínat pouze mezi dvěma regulátory, kdy jeden bude využit pro rozjezd a druhý pro zastavení.

Pro všechny tři metody budeme již uvádět pouze obrázek zrychlení kabiny (popřípadě parametrů PI regulátoru), jelikož u ostatních jsou rozdíly neznatelné oproti těm u GS.

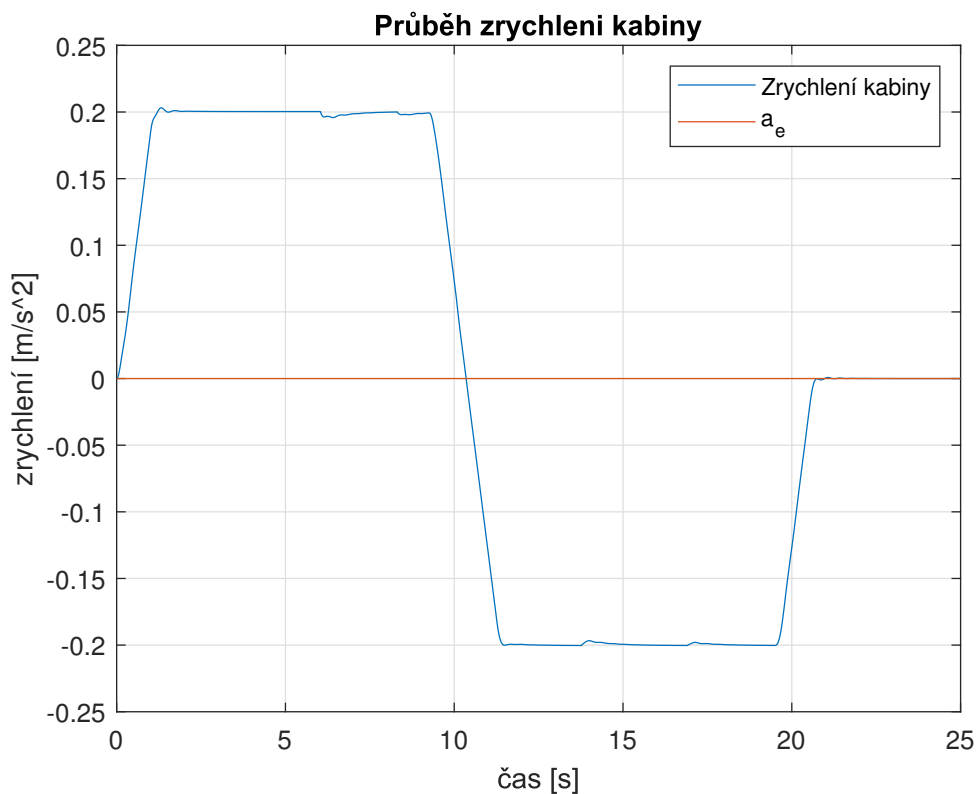
3.1 Selectoring

Tento způsob kmity způsobené změnou regulátorů značně omezí (viz [obrázek 46](#)), ne však zcela. To je způsobené nastavením konstanty vysledování. My všude zvolili $T_t \approx \frac{T_i}{2} = \frac{K_p}{2K_i}$.

Nutno dodat, že jsme si též museli naimplementovat vlastní PI regulátor, jelikož blok PID Controller v Simulinku využívá jakýsi 'Tracking coefficient K_t ' bez specifikování, co tento koeficient znamená (zda se jedná o převrácenou hodnotu konstanty vysledování T_t nebo ne). Po dvoutýdenní anabázi za účelem implementace vysledování výstupu jsme tedy raději vyrobili vlastní PI subsystém (shodný s tím na [obrázku 44](#)), kde můžeme nastavit konstantu $T_t \approx \frac{T_i}{2}$, než se nadále snažit nastavit vhodné K_t metodou pokus omyl.



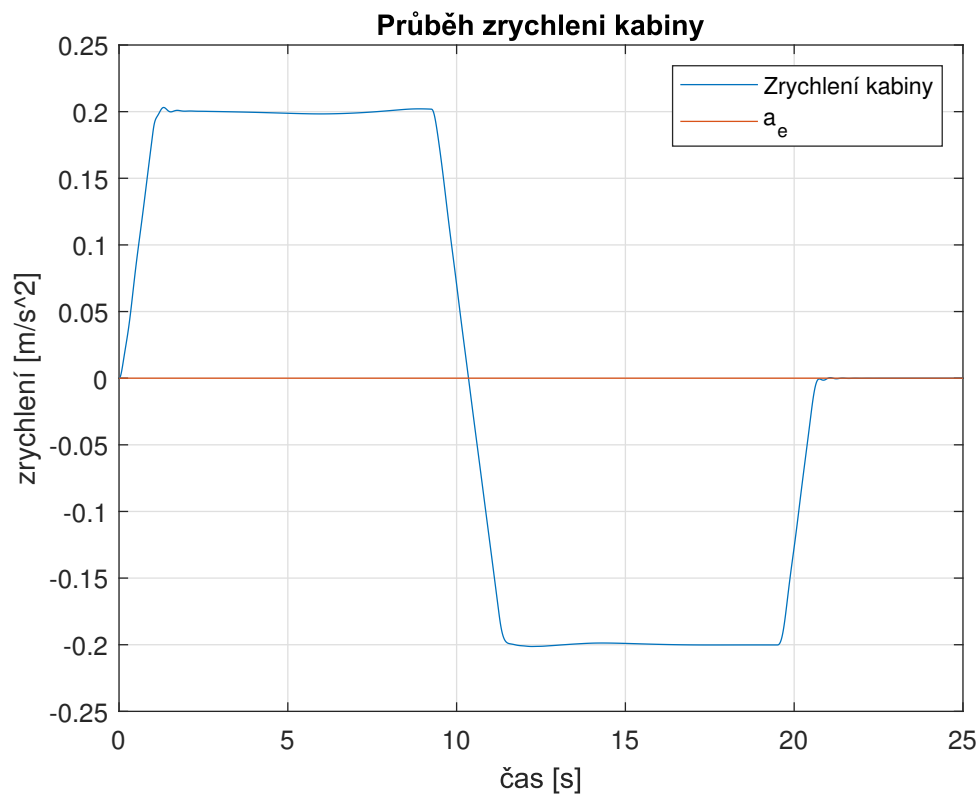
Obrázek 45: Simulační model pro selectoring - detail regulátoru



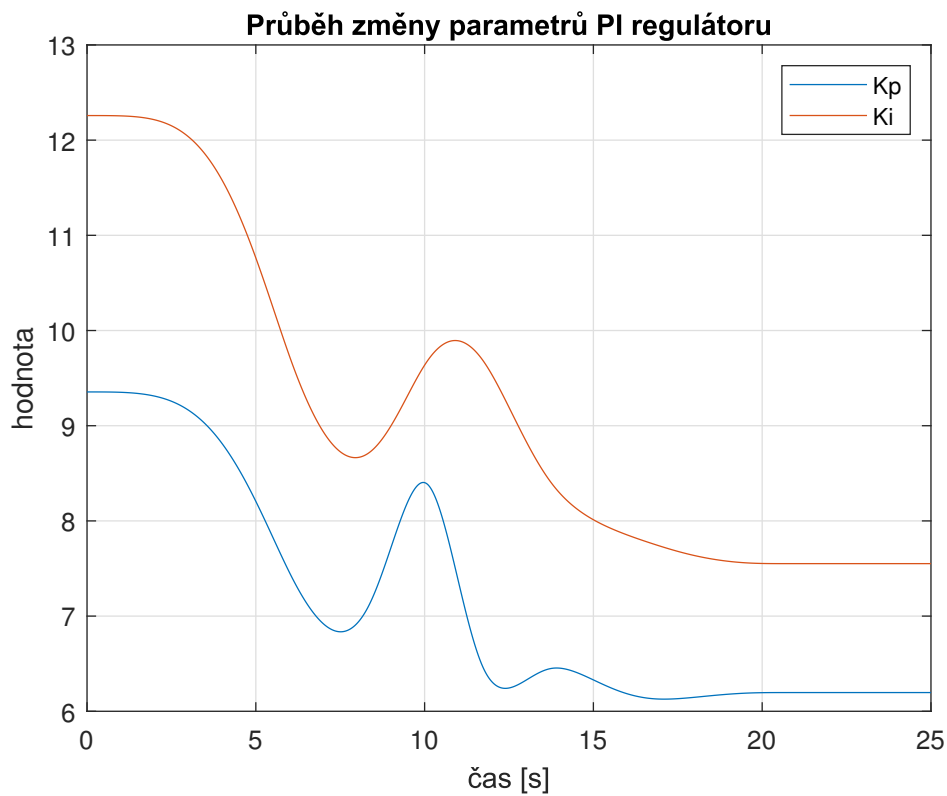
Obrázek 46: Selectoring - zrychlení kabiny

3.2 Proložení K_p a K_i křivkou

Další možností, jak omezit rázy, je nepřepínat K_p a K_i skokově, ale měnit je spojitě. Toho docílíme tím, že v nastavení bloku Lookup Table změním algoritmus interpolační metody z 'Flat' na 'Cubic spline', čili kubickou křivku (schéma stejné jako na [obrázku 38](#)).



Obrázek 47: Proložení parametrů křivkou - zrychlení kabiny



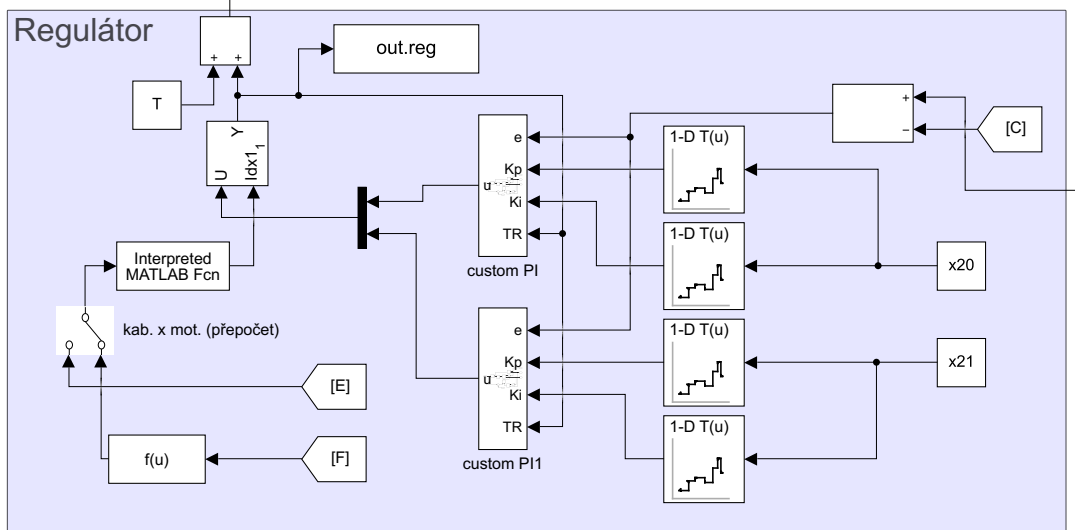
Obrázek 48: Proložení parametrů křivkou - změna parametrů regulátoru

3.3 Dva regulátory

Jak už to tak bývá, nejjednodušší řešení je nejlepší řešení. Tento přístup využívá toho, že předem víme, jaké je startovní a koncové patro. Z množiny všech regulátorů si tedy vybereme dva - nejbližší startovnímu a nejbližší koncovému patru - a mezi nimi se bezrázově přepne uprostřed jízdy. O to se stará následující Matlab skript:

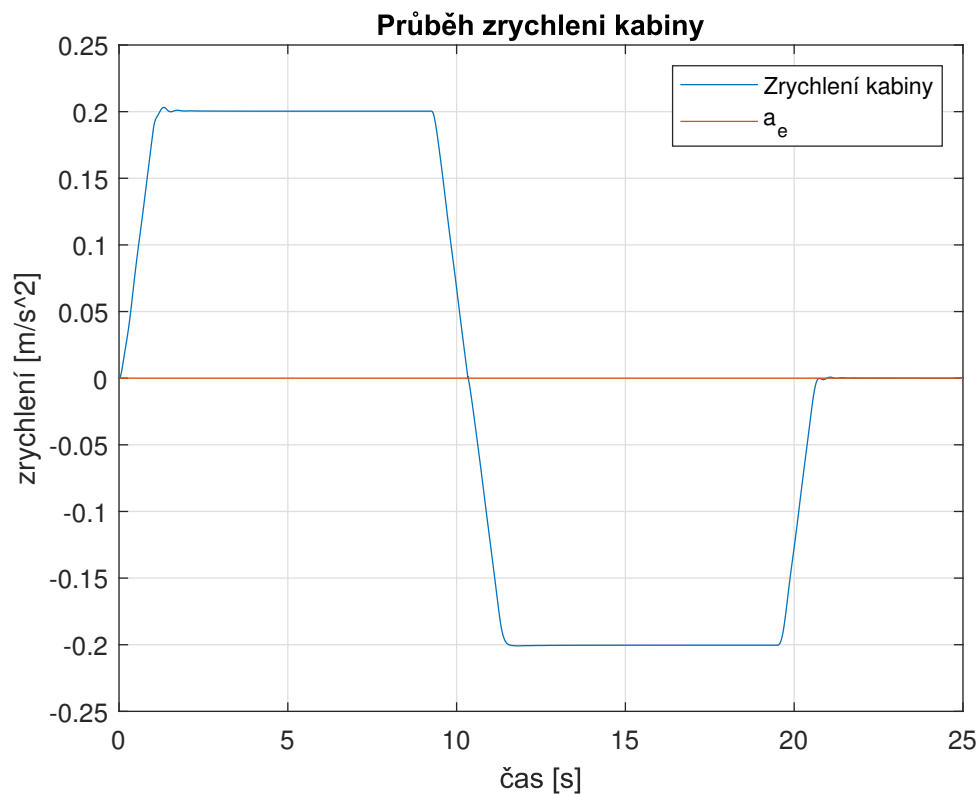
```
1 function [pid] = two_regs_switch_logic(u)
2 x20 = evalin('base', 'x20');
3 x21 = evalin('base', 'x21');
4 mid=(x20+x21)/2;
5
6 if x20>=x21
7     if u>mid
8         pid=1;
9     else
10        pid=2;
11    end
12 else
13    if u<mid
14        pid=1;
15    else
16        pid=2;
17    end
18 end
19 end
```

Simulační schéma je vidět na [obrázku 49](#).



Obrázek 49: Simulační model se dvěma regulátory - detail regulátoru

Výsledná regulace má významně lepší průběh zrychlení kabiny než v případě našeho GS (tz. s nefunkčním sledováním výstupu) nebo selectoringu a dokonce i lepší, než v případě proložení parametrů křivkou - viz [obrázek 50](#). Využíváme při tom opět náš vlastní PI regulátor stejně jako u selectoringu, aby správně fungovalo sledování výstupu.

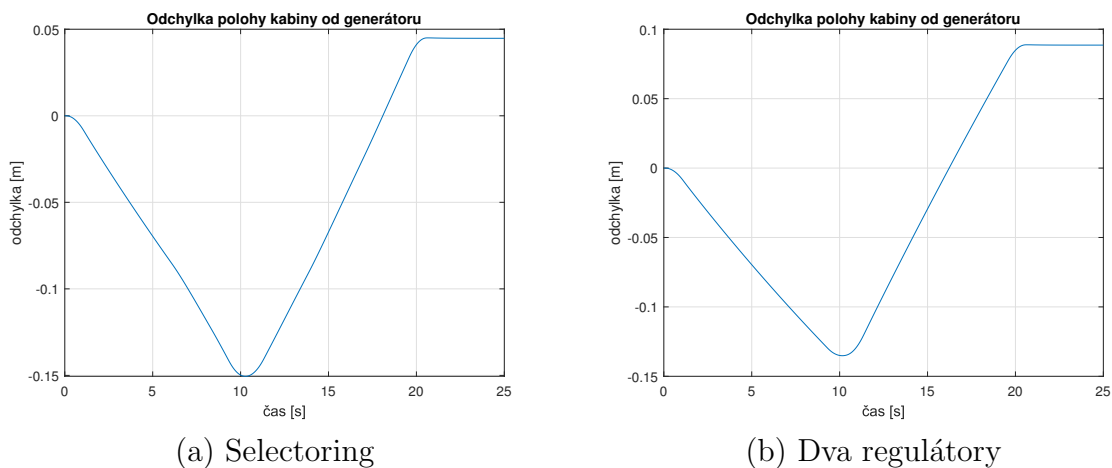


Obrázek 50: Dva regulátory - zrychlení kabiny

3.4 Odchylka polohy

Bohužel se ukazuje, že tento způsob regulace (GS) má zásadní nevýhodu - kvůli regulaci rychlosti je koncová odchylka polohy nenulová. Je to způsobeno tím, že při přepínání regulátorů průběh rychlosti osciluje a ačkoli regulátor tuto odchylku dorovná, tak ona odchylka se naintegruje do polohy, kterou již regulátor nikterak neuvažuje.

Možným řešením by bylo zavedení vnější polohové smyčky.



(a) Selectoring

(b) Dva regulátory

Obrázek 51: Odchylka polohy oproti GT při regulaci GS

4 Linearizace podél trajektorie

Abychom mohli dosáhnout ještě lepší regulace, potřebujeme systém popsat lépe, než jen X systémy linearizovanými v equilibriích (v našem případě jich bylo 10). Toho dosáhneme linearizací systému podél trajektorie, což je způsob, jak vyjádřit dynamiku systému právě v okolí celé jeho požadované pracovní trajektorie namísto jednoho bodu na ní (což je právě linearizace v equilibriu). Popis je pak ve formě časově proměnných matic $\mathbf{A}(t)$, $\mathbf{B}(t)$, tz. jejich složky jsou polynomy proměnné v t (v čase).

Jelikož naše trajektorie je pouze po částech spojitá, tedy popsána v jednotlivých částech jinými polynomy, bylo by zcela obecné analytické řešení linearizace podél trajektorie velmi složité. Tím spíš, že i počet jednotlivých spojitých 'úseků' popsaných jedním polynomem je proměnlivý a závisí na maximálních parametrech $j_{max}, a_{max}, v_{max}$ [7].

My tedy zvolíme následující přístup: Zvolíme jeden konkrétní pohyb výtahu z výšky x_{20} do výšky x_{21} a pro tento pohyb provedeme linearizaci podél trajektorie. Tím dostaneme X = počet úseků matic $\mathbf{A}(t)$, tedy systém popsáný v každé části jednou příslušnou maticí, a pro tyto matice následně zkonstruujeme stavovou zpětnou vazbu $\mathbf{K}(t)$ (buď metodou přiřazení pólů, nebo přes LQR). Pro tuto zpětnou vazbu je samozřejmě potřeba znát stav systému, nikoli pouze výstup motoru, proto ještě před samotnou stavovou zpětnou vazbou budeme muset navrhnout rekonstruktor stavu. To ale předbíháme, v této části se věnujeme linearizaci.

Ještě je dobré dodat, proč musíme linearizovat přímo podél trajektorie a nemůžeme například udělat linearizaci v rovnovážném bodě pro několik tisíc equilibrií (pro všechny body mezi x_{20} a x_{21} s určitým vzorkem), obdobně jako u GS.

Důvodem je, že linearizace v RB uvažuje, že je systém v klidu, tedy že se nepohybuje (má nulovou rychlost). My ale chceme popsat systém takovou časově proměnnou maticí dynamiky $\mathbf{A}(t)$, která uvažuje systém v pohybu s naší požadovanou rychlostí generovanou GT.

Mezi těmito přístupy je tedy rozdíl a vyšly by nám i jiné matice $\mathbf{A}(t)$, tudíž by i navrhovaná stavová zpětná vazba byla navržena špatně, jelikož by při regulaci vždy uvažovala systém v klidu, nikoli v pohybu.

Zvolíme tedy jeden konkrétní pohyb výtahu, abychom měli konstantní počet úseků. V zájmu zachování konzistence zvolíme parametry stejné, jako v [sekci 2.8 \(vztah 32\)](#):

Počáteční poloha kabiny: $x_{20} = 6$ [m]^[4]

Maximální rychlost motoru: 100 [rad · s⁻¹]

Maximální zrychlení motoru: 10 [rad · s⁻²]

Maximální jerk motoru: 10 [rad · s⁻³]

Cílová poloha kabiny: $x_{21} = 25$ [m]

Vzorkování: 0.001 [s]

^[4]Z té již vyplývají ostatní počáteční podmínky - počáteční a koncová poloha kabiny lze jednoduše přepočíst na počáteční a cílovou polohu motoru přes [vztahy uvedené na začátku \(19 až 22\)](#)

Abychom nemuseli popisovat celý (velmi komplikovaný) postup odvození naší trajektorie, znovu uvádíme, že vycházíme z práce [7], ve které je uvedeno, jak přesně rozlišit typ trajektorie (kterých je 6 druhů). Z našich parametrů lze odvodit, že se jedná o trajektorii typu D2 (obrázek 18 vpravo), tudíž v souladu s prací [7] vyjdou (se zaokrouhlením na vzorky):

$$t_j = \dots = 1 ; t_a = \dots = 9.255 ; t_v = t_a + t_j = 10.255$$

z čehož lze vypočítat přepínací časy jednotlivých úseků [s]:

$$t_1 = t_j = 1$$

$$t_2 = t_a = 9.255$$

$$t_3 = t_j + t_a = 10.255$$

$$t_4 = t_v = 10.255$$

$$t_5 = t_j + t_v = 11.255$$

$$t_6 = t_v + t_a = 19.510$$

$$t_7 = t_j + t_v + t_a = 20.510$$

Můžeme vidět, že $t_3 = t_4$, takže ve výsledku budeme mít jen 6 úseků místo maximálních 7. Nyní tedy pro tyto úseky vypíšeme vztahy [7, p. 9] pro $j(t)$, $a(t)$, $v(t)$ a $p(t)$ (pro formalitu ukážeme všech 7, ačkoliv víme, že jeden úsek bude v praxi vynechán). Značkou $p(t)$ rozumíme pozici, značíme jí takto aby značení korespondovalo s prací [7]. Za J budeme uvažovat j_{max} , $j(t)$ pak může nabývat jen hodnot j_{max} , $-j_{max}$, 0 .

Aby na sebe polynomy navazovaly, začínají vždy na hodnotě polynomu z minulé větve. Proto zavádíme pomocné proměnné a_i, v_i, p_i . Dále se ještě kvůli možnosti přeskočení některého z úseků zavádí pomocné proměnné $a_{i+1}, v_{i+1}, p_{i+1}$, které jsou pak využity v úseku po následujícím (současný úsek + 2), protože následující byl přeskočen.

Po detailním prostudování rovnic [7, p. 9] je význam pomocných proměnných jasný, jeho popsání je však poměrně krkolomné. Dále je náš generátor trajektorie (vytvořen v projektu [1] kolegou Tvrzem) ještě lehce upraven, protože nestrartuje vždy z nuly, ale z p_0 , a také protože generuje trajektorii oběma směry, proto zavádíme ještě $d = \pm 1$ pro označení směru jízdy.

Zmíněná p_0 je obecná startovní poloha, protože generátor trajektorie generuje trajektorii pro obecný pohyb po S-křivce, bez ohledu na to, zda jde o pohyb lineární nebo rotační.

Jelikož náš generátor trajektorie je využíván pro generování trajektorie pro motor, je jeho počáteční polohou počáteční poloha motoru, čili $p_0 = \phi_0$. Proměnná p_{final} pak značí koncovou polohu, pro nás tedy koncová poloha motoru, tudíž $p_{final} = \phi_1$, kterou získáme přepočtem ze žádané x_{21} .

Nyní tedy uvedeme obecné vzorce trajektorie (rovnice 34), odvozené z práce [7] ale upravené pro generování trajektorie oběma směry:

$$p_0 < p_{final} :$$

$$d = 1$$

$$p_0 > p_{final} :$$

$$d = -1$$

$$t \in [0; t_1) :$$

$$j(t) = j_{max}$$

$$a(t) = j_{max} \cdot t$$

$$a_2 = a_1 = a(t_1)$$

$$v(t) = j_{max} \cdot \frac{t^2}{2}$$

$$v_2 = v_1 = v(t_1)$$

$$p(t) = p_0 + \left(j_{max} \cdot \frac{t^3}{6} \right) \cdot d$$

$$p_2 = p_1 = p(t_1)$$

$$t \in [t_1; t_2) :$$

$$j(t) = 0$$

$$a(t) = a_{max}$$

$$a_3 = a_2 = a(t_2)$$

$$v(t) = v_1 + a_{max} \cdot (t - t_1)$$

$$v_3 = v_2 = v(t_2)$$

$$p(t) = p_1 + \left(v_1 \cdot (t - t_1) + a_{max} \cdot \frac{(t - t_1)^2}{2} \right) \cdot d$$

$$p_3 = p_2 = p(t_2)$$

$$t \in [t_2; t_3) :$$

$$j(t) = -j_{max}$$

$$a(t) = a_2 - j_{max} \cdot (t - t_2)$$

$$a_4 = a_3 = a(t_3)$$

$$v(t) = v_2 + a_2 \cdot (t - t_2) - j_{max} \cdot \frac{(t - t_2)^2}{2}$$

$$v_4 = v_3 = v(t_3)$$

$$p(t) = p_2 + \left(v_2 \cdot (t - t_2) + a_2 \cdot \frac{(t - t_2)^2}{2} - j_{max} \cdot \frac{(t - t_2)^3}{6} \right) \cdot d$$

$$p_4 = p_3 = p(t_3)$$

$$t \in [t_3; t_4) :$$

$$j(t) = 0$$

$$a(t) = 0$$

$$a_5 = a_4 = a(t_4)$$

$$v(t) = v_3 + a_{max} \cdot (t - t_3)$$

$$v_5 = v_4 = v(t_4)$$

$$p(t) = p_3 + (v_3 \cdot (t - t_3)) \cdot d$$

$$p_5 = p_4 = p(t_4)$$

$$t \in [t_4; t_5) :$$

$$j(t) = -j_{max}$$

$$a(t) = -j_{max} \cdot (t - t_4)$$

$$a_6 = a_5 = a(t_5)$$

$$v(t) = v_4 - j_{max} \cdot \frac{(t - t_4)^2}{2}$$

$$v_6 = v_5 = v(t_5)$$

$$p(t) = p_4 + \left(v_4 \cdot (t - t_4) - j_{max} \cdot \frac{(t - t_4)^3}{6} \right) \cdot d$$

$$p_6 = p_5 = p(t_5)$$

$$\begin{aligned}
t \in [t_5; t_6) : \\
j(t) &= 0 \\
a(t) &= -a_{max} & a_6 &= a(t_6) \\
v(t) &= v_5 - a_{max} \cdot (t - t_5) & v_6 &= v(t_6) \\
p(t) &= p_5 + \left(v_5 \cdot (t - t_5) - a_{max} \cdot \frac{(t - t_5)^2}{2} \right) \cdot d & p_6 &= p(t_6) \\
t \in [t_6; t_7) : \\
j(t) &= j_{max} \\
a(t) &= a_6 + j_{max} \cdot (t - t_6) \\
v(t) &= v_6 + a_6(t - t_6) + j_{max} \cdot \frac{(t - t_6)^2}{2} \\
p(t) &= p_6 + \left(v_6 \cdot (t - t_6) + a_6 \cdot \frac{(t - t_6)^2}{2} + j_{max} \cdot \frac{(t - t_6)^3}{6} \right) \cdot d \\
t \geq t_7 : \\
j(t) &= 0 \\
a(t) &= 0 \\
v(t) &= 0 \\
p(t) &= p_{final} \\
\forall t : \\
j(t) &= d \cdot j(t) \\
a(t) &= d \cdot a(t) \\
v(t) &= d \cdot v(t)
\end{aligned} \tag{34}$$

Nyní můžeme vypočítat trajektorie pro motor, kabinu a protizávaží (každá část systému má vlastní trajektorii a kvůli nelinearitě nelze polynomy snadno vzájemně přepočítat). Protože výpočty jsou velmi komplikované (řetěží se), vytvořili jsme automatický skript pro výpočet potřebných polynomů. Bohužel se ukazuje, že různé jsou nejen parametry j_{max} , a_{max} , v_{max} a polynomy, ale i přepínací časy. To je způsobeno rovněž nelinearitou, protože změna výšky lana (rozdíl jeho počáteční a koncové délky) bude v principu menší než změna výšky výtahu (čím delší lano je, tím více se prodlouží, nelineárně), přičemž poloha motoru je lineárně převoditelná právě na délku lana (l_2), nikoli na výšku kabiny (x_2). Nyní tedy uvedeme hodnoty pro jednotlivé komponenty - [tabulka vztahů 35](#) (zokrouhleně, pro přehlednost):

Protože generátor trajektorie generuje trajektorii pro motor a nikoli pro kabinu, je nutno parametry j_{max} , a_{max} , v_{max} vynásobit virtuálním poloměrem $r = \frac{r_n}{n} = \frac{0.3}{15} = \frac{1}{50} = 0.02$ (poloměr s uvažováním převodu, de facto tedy konstanta určující rozdíl mezi rotační rychlostí motoru a posuvnou rychlostí kabiny). Tudíž $j_{max} = 0.2ms^{-3}$, $a_{max} = 0.2ms^{-2}$, $v_{max} = 2ms^{-1}$.

Dále uvažujeme, že $p_0 = x_{20} = 6m$ a $p_{final} = x_{21} = 25m$.

Motor ($\phi, \dot{\phi}, \ddot{\phi}$)	Kabina ($x_2, \dot{x}_2, \ddot{x}_2$)	Protizávaží ($x_1, \dot{x}_1, \ddot{x}_1$)
$p_0 = \phi_0 \doteq 1350.34$	$p_0 = x_{20} = 6$	$p_0 = x_{10} \doteq 27$
$p_{fin.} = \phi_1 \doteq 401.42$	$p_{fin.} = x_{21} = 25$	$p_{fin.} = x_{11} \doteq 7.98$
$j_{max} = 10$	$j_{max} = 0.2$	$j_{max} = 0.2$
$a_{max} = 10$	$a_{max} = 0.2$	$a_{max} = 0.2$
$v_{max} = 10$	$v_{max} = 0.2$	$v_{max} = 0.2$
$t_1 = 1$	$t_1 = 1$	$t_1 = 1$
$t_2 = 9.255$	$t_2 = 9.26$	$t_2 = 9.264$
$t_3 = 10.255$	$t_3 = 10.26$	$t_3 = 10.264$
$t_4 = 10.255$	$t_4 = 10.26$	$t_4 = 10.264$
$t_5 = 11.255$	$t_5 = 11.26$	$t_5 = 11.264$
$t_6 = 19.510$	$t_6 = 19.52$	$t_6 = 19.528$
$t_7 = 20.510$	$t_7 = 20.52$	$t_7 = 20.528$

(35)

Nyní tedy budeme uvažovat naše konkrétní hodnoty.

Protože $x_{20} < x_{21}$, tak bude $d = 1$. Naopak pro protizávaží, které jede opačným směrem než kabina (vzhledem ke zvolené soustavě značení) bude směr $d = -1$. Tudíž i pro motor bude směr $d = -1$, protože i ten se otáčí opačným směrem než kabina (vzhledem ke zvolené soustavě).

Po dosažení $j_{max}, a_{max}, t_1, t_2, t_3, t_4, t_5, t_6, t_7$ (v_{max} se využívá pouze k určení typu trajektorie, ve výše uvedených polynomech nefiguruje) nám tedy vyjdou polynomy proměnné v čase příslušné pohybu dané komponenty systému (kabina - [soustava rovnic 36](#)/protizávaží - [soustava rovnic 37](#)/motor - [soustava rovnic 38](#)).

Jelikož výpočty jsou numerické, nejsou uvedené polynomy zcela přesné, navíc je zde záměrně zaokrouhlujeme, opět v zájmu přehlednosti. Samotný skript provádí jednotlivé kroky linearizace ihned za sebou a tudíž nezaokrouhluje, dochází tedy pouze k chybám vlivem omezené přesnosti numeriky.

Trajektorie kabiny - $x_{2p} = p(t), \dot{x}_{2p} = v(t), \ddot{x}_{2p} = a(t)$

$t \in [0; t_1) :$

$$\begin{aligned}j(t) &= 0.2 \\a(t) &= 0.2t \\v(t) &= 0.1t^2 \\p(t) &= 0.333t^3 + 6\end{aligned}$$

$t \in [t_1; t_2) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 0.2 \\v(t) &= 0.2t - 0.1 \\p(t) &= 0.1t^2 - 0.1t + 6.03\end{aligned}$$

$t \in [t_2; t_3) :$

$$\begin{aligned}j(t) &= -0.2 \\a(t) &= -0.2t + 2.05 \\v(t) &= -0.1t^2 + 2.05t - 8.67 \\p(t) &= -0.0333t^3 + 1.03t^2 - 8.67t + 32.5\end{aligned}$$

$t \in [t_3; t_4) :$

úsek je vynechán

$t \in [t_4; t_5) :$

$$\begin{aligned}j(t) &= -0.2 \\a(t) &= -0.2t + 2.05 \\v(t) &= -0.1t^2 + 2.05t - 8.67 \\p(t) &= -0.0333t^3 + 1.03t^2 - 8.67t + 32.5\end{aligned}$$

$t \in [t_5; t_6) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= -0.2 \\v(t) &= -0.2t + 4 \\p(t) &= -0.1t^2 + 4t - 15.1\end{aligned}$$

$t \in [t_6; t_7) :$

$$\begin{aligned}j(t) &= 0.2 \\a(t) &= 0.2t - 4.1 \\v(t) &= 0.1t^2 - 4.1t + 42.1 \\p(t) &= 0.0333t^3 - 2.05t^2 + 42.1t - 263.0\end{aligned}$$

$t \geq t_7 :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 0 \\v(t) &= 0 \\p(t) &= 25\end{aligned}$$

(36)

Trajektorie protizávaží - $x_{1p} = p(t), \dot{x}_{1p} = v(t), \ddot{x}_{1p} = a(t)$

$t \in [0; t_1) :$

$$\begin{aligned}j(t) &= -0.2 \\a(t) &= -0.2t \\v(t) &= -0.1t^2 \\p(t) &= -0.333t^3 + 27\end{aligned}$$

$t \in [t_1; t_2) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= -0.2 \\v(t) &= -0.2t + 0.1 \\p(t) &= -0.1t^2 + 0.1t + 27\end{aligned}$$

$t \in [t_2; t_3) :$

$$\begin{aligned}j(t) &= 0.2 \\a(t) &= 0.2t - 2.05 \\v(t) &= 0.1t^2 - 2.05t + 8.68 \\p(t) &= 0.0333t^3 - 1.03t^2 + 8.68t + 0.464\end{aligned}$$

$t \in [t_3; t_4) :$

úsek je vynechán

$t \in [t_4; t_5) :$

$$\begin{aligned}j(t) &= 0.2 \\a(t) &= 0.2t - 2.05 \\v(t) &= 0.1t^2 - 2.05t + 8.68 \\p(t) &= 0.0333t^3 - 1.03t^2 + 8.68t + 0.464\end{aligned}$$

$t \in [t_5; t_6) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 0.2 \\v(t) &= 0.2t - 4.01 \\p(t) &= 0.1t^2 - 4.01t + 48.1\end{aligned}$$

$t \in [t_6; t_7) :$

$$\begin{aligned}j(t) &= -0.2 \\a(t) &= -0.2t + 4.11 \\v(t) &= -0.1t^2 + 4.11t - 42.1 \\p(t) &= -0.0333t^3 + 2.05t^2 - 42.1t + 296.0\end{aligned}$$

$t \geq t_7 :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 0 \\v(t) &= 0 \\p(t) &= 7.98\end{aligned}$$

(37)

Trajektorie motoru - $\phi_p = p(t), \dot{\phi}_p = v(t), \ddot{\phi}_p = a(t)$

$t \in [0; t_1) :$

$$\begin{aligned}j(t) &= -10 \\a(t) &= -10t \\v(t) &= -5t^2 \\p(t) &= -1.67t^3 + 1350.34\end{aligned}$$

$t \in [t_1; t_2) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= -10 \\v(t) &= -10t + 5 \\p(t) &= -5t^2 + 5t + 1348.67\end{aligned}$$

$t \in [t_2; t_3) :$

$$\begin{aligned}j(t) &= 10 \\a(t) &= 10t - 102.5 \\v(t) &= 5t^2 - 102.5t + 433.2 \\p(t) &= 1.67t^3 - 51.265t^2 + 433.2t + 27.7\end{aligned}$$

$t \in [t_3; t_4) :$

úsek je vynechán

$t \in [t_4; t_5) :$

$$\begin{aligned}j(t) &= 10 \\a(t) &= 10t - 102.5 \\v(t) &= 5t^2 - 102.5t + 433.2 \\p(t) &= 1.67t^3 - 51.265t^2 + 433.2t + 27.7\end{aligned}$$

$t \in [t_5; t_6) :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 10 \\v(t) &= 10t - 200 \\p(t) &= 5t^2 - 200t + 2403.5\end{aligned}$$

$t \in [t_6; t_7) :$

$$\begin{aligned}j(t) &= -10 \\a(t) &= -10t + 205 \\v(t) &= -5t^2 + 205t - 2103 \\p(t) &= -1.67t^3 + 102.5t^2 - 2103t + 14778\end{aligned}$$

$t \geq t_7 :$

$$\begin{aligned}j(t) &= 0 \\a(t) &= 0 \\v(t) &= 0 \\p(t) &= 401.42\end{aligned}$$

(38)

Můžeme vidět, že polynomy jsou skutečně rozdílné a vzájemně je nelze snadno přepočítat.

Nyní tedy konečně můžeme přistoupit k samotné linearizaci. Pracovní trajektorii $\mathbf{z}_p = [z_{1p}, z_{2p}, z_{3p}, z_{4p}, z_{5p}, z_{6p}] = [x_{1p}, x_{2p}, \phi_p, \dot{x}_{1p}, \dot{x}_{2p}, \dot{\phi}_p]$; $u = T$ nyní využijeme pro linearizaci podél trajektorie. Nebudeme zde odvozovat to, jak samotná linearizace podél trajektorie probíhá. Jejím konečným krokem však je, že se do matic naší obecné linearizace z bodu 2.3 dosadí místo $x_{20}, l_{20}, x_{10}, l_{10}, \phi_0$ právě pracovní trajektorie \mathbf{z}_p , kterážto není tvořena konstantami, ale časově proměnnými polynomy. Díky tomu pak dostaneme časově variantní matici $\mathbf{A}(t)$. Stavů pak nejsou tvořeny konkrétními hodnotami x_2, x_1, \dots , ale jejich odchylkami od pracovní trajektorie, jedná se totiž o odchylkovou aproximaci:

$$\begin{aligned}\Delta \dot{\mathbf{z}}(t) &= \left. \frac{\partial \mathbf{f}}{\partial \mathbf{z}} \right|_{\mathbf{z}_p(t)} \Delta \mathbf{z}(t) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right|_{\mathbf{z}_p(t)} \Delta \mathbf{u}(t) \\ \Delta \mathbf{z}(t) &= \mathbf{z}(t) - \mathbf{z}_p(t) \\ \Delta \dot{\mathbf{z}}(t) &= \dot{\mathbf{z}}(t) - \dot{\mathbf{z}}_p(t)\end{aligned}\tag{39}$$

Uvažujeme tedy naši obecnou matici \mathbf{A} (vztah 18):

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}} = \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_0}{m_1 l_1} & 0 & \frac{k_0 r z_1}{m_1 l_1^2} & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \frac{-k_0}{m_2 l_2} & \frac{-k_0 r z_2}{m_2 l_2^2} & 0 & \frac{-b_0}{m_2 l_2} & \frac{-b_0 r}{m_2 l_2} \\ \frac{k_0 r}{J l_1} & \frac{-k_0 r}{J l_2} & \frac{-k_0 r^2 (z_2 l_1^2 + z_1 l_2^2)}{J l_1^2 l_2^2} & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2 (l_1 + l_2) b_0 - b_f l_1 l_2}{J l_1 l_2} \end{bmatrix}$$

Nyní zjišťujeme, že jelikož je matice zcela obecná a obsahuje nejen stavy z_1, z_2 , ale i délky lan l_1, l_2 , musíme provést linearizaci podél trajektorie i pro lana (získat polynomy určující změny jejich délek v čase z počátečních l_{10}, l_{20} do cílových l_{11}, l_{21}), protože v případě linearizace v RB se za ně dosadí konstantní hodnoty (čísla) vypočtená z počátečních podmínek, zatímco zde se délky lan mění společně s polohami kabiny a protizávaží. Nepovažujeme za nutné zde tyto trajektorie uvádět v polynomiální podobě jako výše pro kabinu atd., protože konkrétní hodnoty se stejně budou vždy lišit v závislosti na počáteční a cílové poloze (u výše uvedených trajektorií jsme chtěli hlavně ukázat, že polynomy nejsou vzájemně triviálně převoditelné), navíc máme vytvořený automatický skript, který tyto polynomy rychle spočítá pro jakékoli zadané parametry.

Do naší matice nyní dosadíme naše polynomy pracovní trajektorie $z_{1p} = x_{1p}, z_{2p} = x_{2p}$ a polynomy změn délky lan l_1, l_2 , čímž dostaneme naši časově variantní matici $\mathbf{A}_p(t)$ která je linearizovaná podél pracovní trajektorie \mathbf{z}_p :

$$\mathbf{A}_p(t) = \mathbf{A}|_{\mathbf{z}_p(t)} = \begin{bmatrix} \text{polynom}(t) & \text{polynom}(t) & \dots \\ \text{polynom}(t) & \text{polynom}(t) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}_{(6 \times 6)}\tag{40}$$

Protože je nesmysl mít rozdílné přepínací časy pro jednotlivé trajektorie, budeme pro všechny polynomy používat pouze přepínací časy kabiny. Díky tomu dostaneme jen 8 matic $\mathbf{A}(t)$ místo ± 22 . Těchto osm polynomiálních matic nám nyní zcela popisuje systém linearizovaný podél trajektorie.

Pro účely simulace pak do těchto matic dosadíme za čas $t = 0 : 0.001 : 25$, přičemž vždy dosazujeme s respektem k přepínacím časům do konkrétní matice:

```

1 t_=0:vzorek:25;
2 for i=1:length(t_)
3     t=t_(i);
4     if t <= t1
5         A_p(:, :, i)=eval(A1);
6     elseif t1 < t && t <=t2
7         A_p(:, :, i)=eval(A2);
8     elseif t2 < t && t <=t3
9         A_p(:, :, i)=eval(A3);
10    elseif t3 < t && t <=t4
11        A_p(:, :, i)=eval(A4);
12    elseif t4 < t && t <=t5
13        A_p(:, :, i)=eval(A5);
14    elseif t5 < t && t <=t6
15        A_p(:, :, i)=eval(A6);
16    elseif t > t6 && t < t7
17        A_p(:, :, i)=eval(A7);
18    elseif t >= t7
19        A_p(:, :, i)=eval(A8);
20    end
21    LKT(:, :, i) = ss(A_p(:, :, i),B,C,D);
22 end

```

LKT (Lin. Kolem Trajektorie) představuje LTV (Linear Time Varying) systém, který potřebujeme pro vyzkoušení regulace na linearizovaném systému předtím, než se posuneme na nelineární model.

5 Časově proměnný lineárně kvadratický regulátor

Nyní máme tedy systém linearizovaný podél trajektorie a můžeme se vrhnout na návrh LQR. To je regulátor, který pro zadaný systém (systém se stavovou zpětnou vazbou):

$$\begin{aligned} \Delta \dot{\mathbf{z}}(t) &= \mathbf{A}(t)|_{z_p(t)} \Delta \mathbf{z}(t) + \mathbf{B}(t)|_{z_p(t)} \Delta \mathbf{u}(t), \quad \Delta \mathbf{z}(0) = \Delta \mathbf{z}_0 \\ \Delta \mathbf{u}(t) &= -\mathbf{K} \Delta \mathbf{z}(t) \end{aligned} \quad (41)$$

splňuje kritérium:

$$I = \int_0^{\infty} e^2(t) dt = \int_0^{\infty} (\Delta \mathbf{z}(t)^T \mathbf{Q} \Delta \mathbf{z}(t) + \Delta \mathbf{u}(t)^T \mathbf{R} \Delta \mathbf{u}(t)) dt \rightarrow \min. \quad (42)$$

tedy že svým řízením minimalizuje kvadrát odchylky systému od požadované hodnoty^[5] (v našem případě trajektorie^[6]).

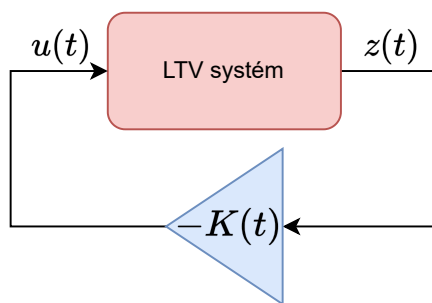
Jeho návrh spočívá v zadání matic \mathbf{Q} a \mathbf{R} které popisují váhu stavů, tedy v podstatě to, jak důležitý pro nás stav je vzhledem k regulaci. Z těchto matic je pak vypočtena stavová zpětná vazba $\Delta \mathbf{u}(t) = -\mathbf{K} \Delta \mathbf{z}(t)$. Je důležité, aby byla $\mathbf{Q} \geq 0$ (pozitivně semidefinitní) a $\mathbf{R} > 0$ (pozitivně definitní). Obě matice je vhodné volit diagonální.

^[5]KKY/NŘS

^[6]Proto jsou v rovnicích odchylky Δ , viz [vztah 39](#)

Jelikož náš systém je LTV (časově proměnný) musíme navrhnout i časově proměnný LQR, standardně označovaný jako SDRE[9], tzn. vyjde nám časově variantní stavová zpětná vazba $\mathbf{K}(t)$. Při jeho návrhu budeme navíc uvažovat, že máme k dispozici celý stav systému a rekonstruktor navrhne až poté (podle separačního teorému lze návrh těchto dvou částí oddělit^[7]). Následně je spojíme, čímž získáme tzv. dynamický kompenzátor.

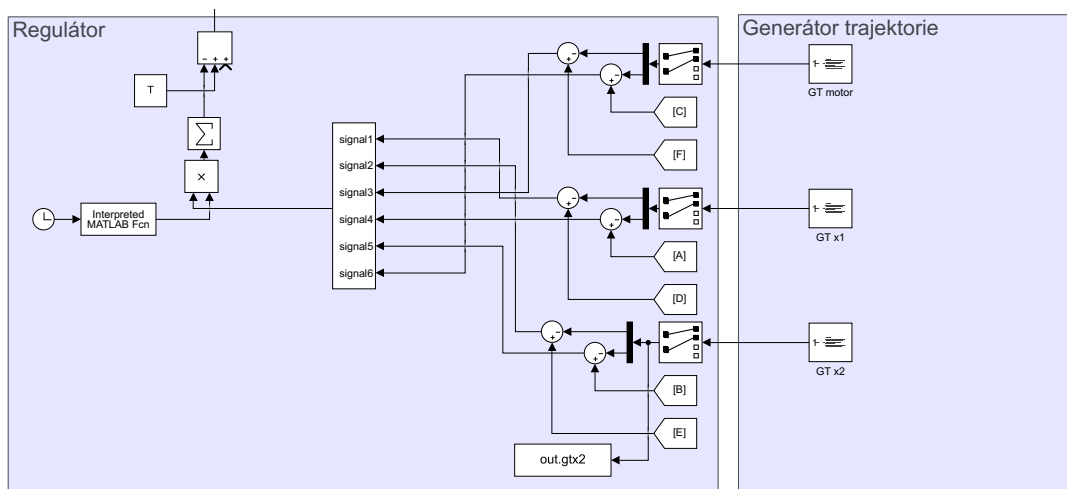
Struktura systému je pak následující (obrázek 52):



Obrázek 52: Princip (časově proměnné) LQR regulace

V případě regulování linearizovaného systému (například v bloku *LPV system*) by výstupem tohoto systému byly odchylky od trajektorie (podél které byl linearizován) a regulace by tedy spočívala jen v zavedení stavové zpětné vazby $\mathbf{u}(t) = -\mathbf{K}\mathbf{z}(t)$, která by systém převedla z počátečních podmínek do počátku (do nuly).

Naše schéma však vypadá takto (obrázek 56):



Obrázek 53: Schéma SDRE regulace

My totiž regulujeme nelineární systém, jehož výstupem je opravdový stav systému, nikoli odchylka. Od těchto stavů tedy musíme odečíst požadovanou trajektorii (kterou získáme z generátoru trajektorie), čímž získáme odchylku, kterou již můžeme použít pro stavovou regulaci získanou řešením SDRE.

^[7]KKY/LS2

Tato regulace by řídila systém tak, že by tlačila odchylku systému od požadované pracovní trajektorie ($\Delta \mathbf{z}(t) = \mathbf{z}(t) - \mathbf{z}_p(t)$) k nule, systém by tedy zhruba kopíroval žádanou trajektorii (to si můžeme ověřit z výsledků simulací - obrázky 54 a 55).

Poněvadž neexistuje obecný návod, jak volit matice \mathbf{Q} a \mathbf{R} , budeme postupovat takto: Protože chceme regulovat hlavně pozici a především rychlost kabiny, zvolíme pro tyto stavy (x_2, \dot{x}_2) výrazně vyšší hodnoty v matici \mathbf{Q} , než pro ty ostatní. Matici \mathbf{R} pak zvolíme jednotkovou. Následně postupujeme metodou pokus-omyl a zkusíme upravovat hodnoty matice \mathbf{Q} tak, aby se vlastnosti regulace zlepšovaly. Významně nám v tomto pomohla znalost vnitřní dynamiky systému, proto jsme již po pár pokusech dostali ideální matici \mathbf{Q} :

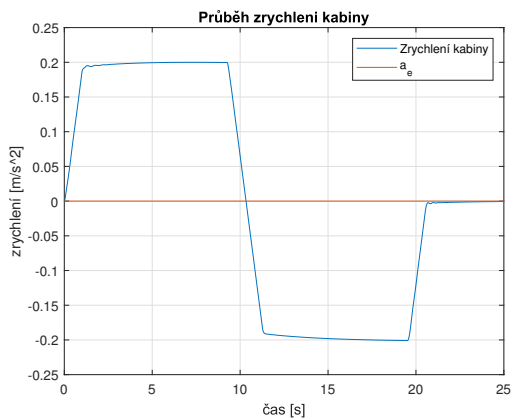
$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 \times 10^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 \times 10^5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{R} = \mathbf{1} \quad (43)$$

Matice není nutno volit časově proměnné, časovou proměnnost LSZV $\mathbf{K}(t)$ zajistí to, že je proměnná matice $\mathbf{A}(t)$. Navíc, změna parametrů matice \mathbf{Q} by znamenala změnu výstupu regulátoru, což by způsobilo rázy, stejně jako v případě GS.

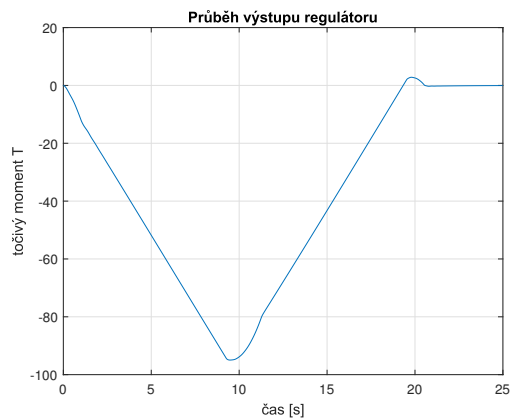
Simulaci tedy provádíme vždy dvakrát - směrem nahoru i dolů, abychom otestovali, že regulátor funguje oběma směry. To lze jen díky tomu, že jsme si vytvořili automatický skript pro linearizaci podél trajektorie, jak jsme již zmínili dříve, protože je potřeba vždy vytvořit novou matici \mathbf{A}_p (příslušnou dané trajektorii která je směrem nahoru jiná než směrem dolů).

Nejdříve tedy simulace směrem dolů - [obrázek 54](#).

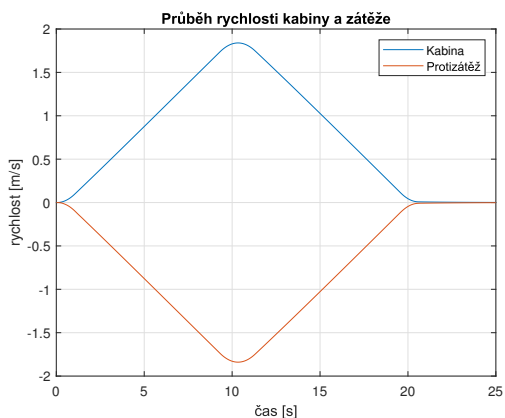
Následně pak simulace směrem nahoru - [obrázek 55](#).



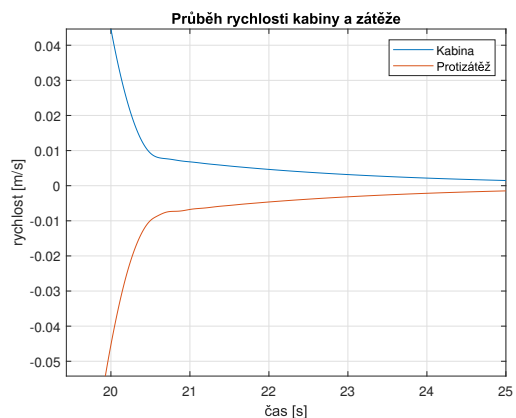
(a) Zrychlení kabiny



(b) Výstup regulátoru



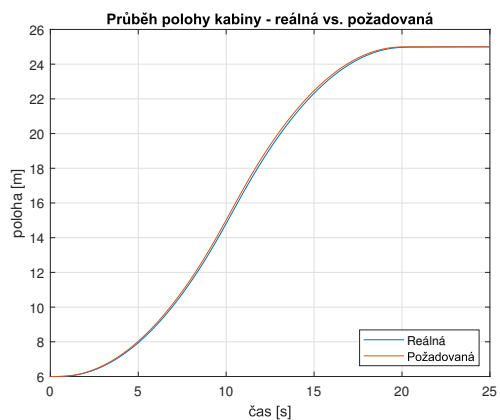
(c) Rychlost kabiny a protizátěže



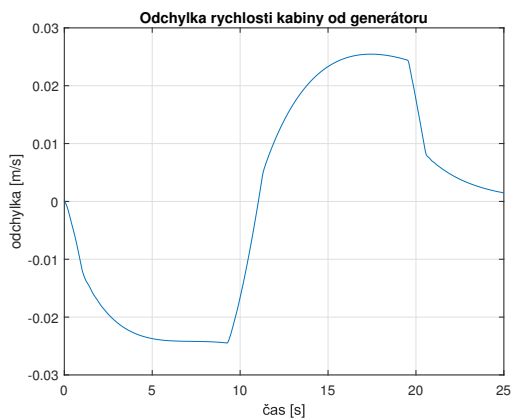
(d) Rychlost kabiny a protizátěže - detail



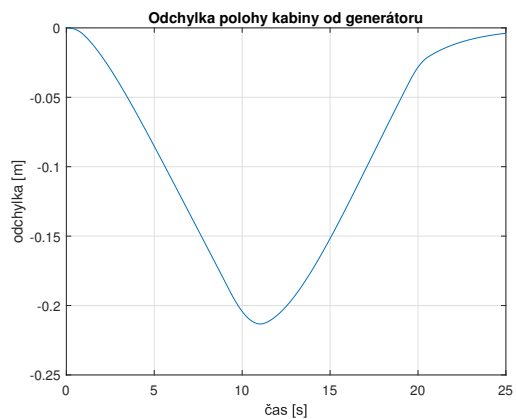
(e) Rychlost kabiny pož. vs. reálná



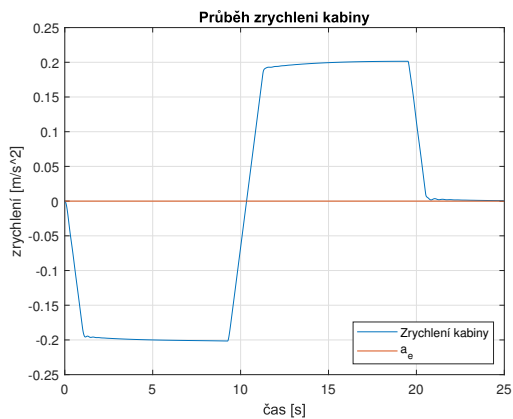
(f) Poloha kabiny pož. vs. reálná



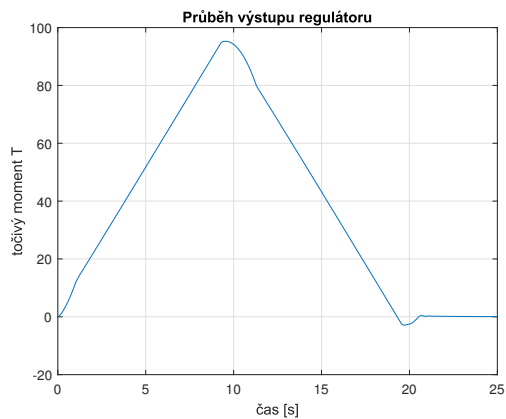
(g) Odchylka rychlosti kabiny od pož.



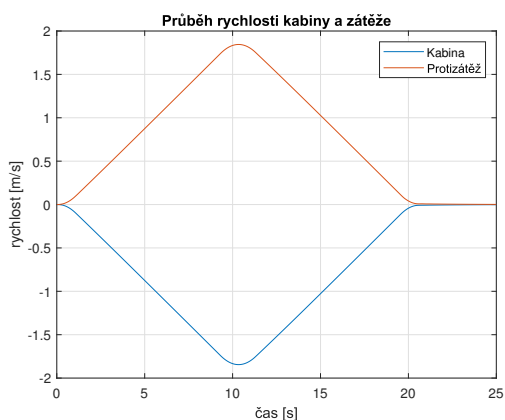
(h) Odchylka polohy kabiny od pož.



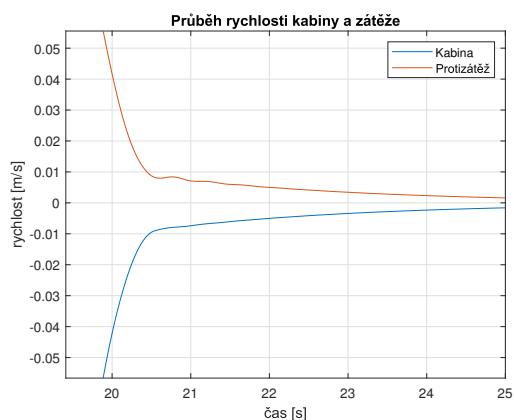
(a) Zrychlení kabiny



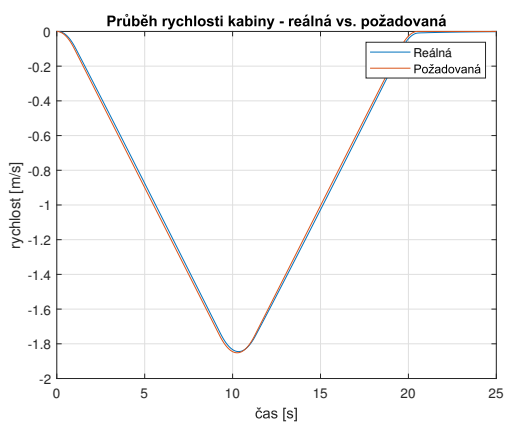
(b) Výstup regulátoru



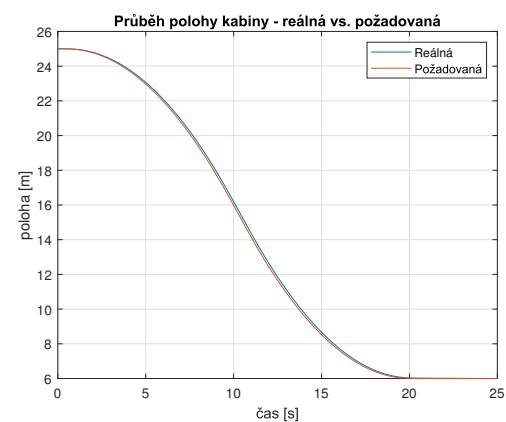
(c) Rychlost kabiny a protizátěže



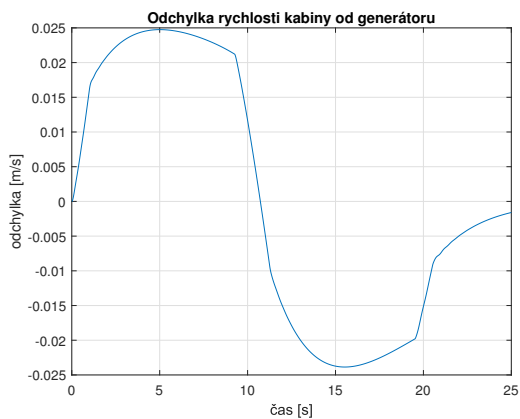
(d) Rychlost kabiny a protizátěže - detail



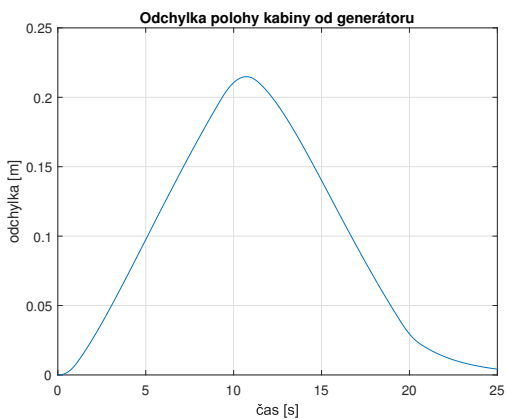
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná

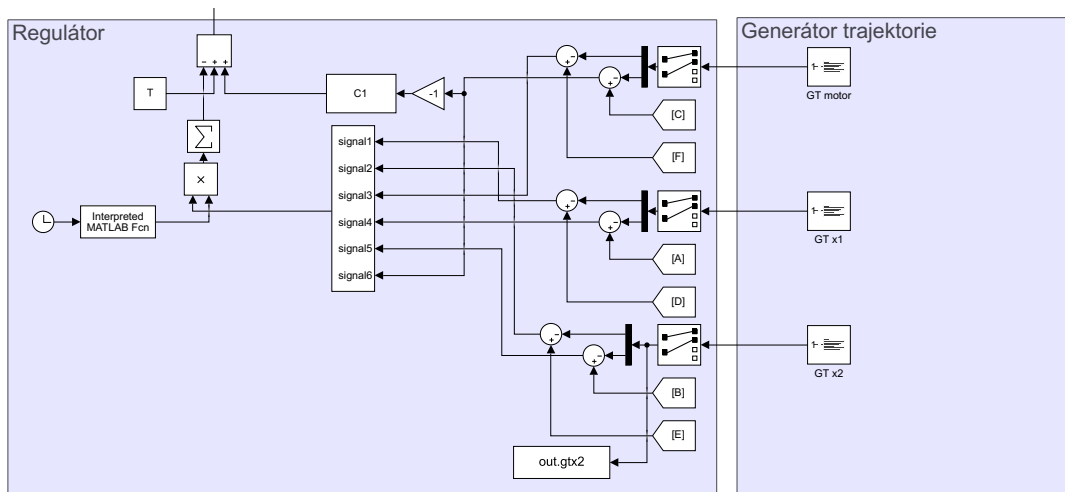


(g) Odchylka rychlosti kabiny od pož.



(h) Odchylka polohy kabiny od pož.

Za výhodný považujeme hlavně fakt, že posledních několik vteřin dojíždí kabina velmi pomalu a to bez přepnutí regulátoru (což je přístup v praxi často využívaný). Výtah tedy pomalu dojede na koncové spínače (umístěné v každém patře), které jeho pohyb zastaví. Dále budeme uvažovat následující schéma:

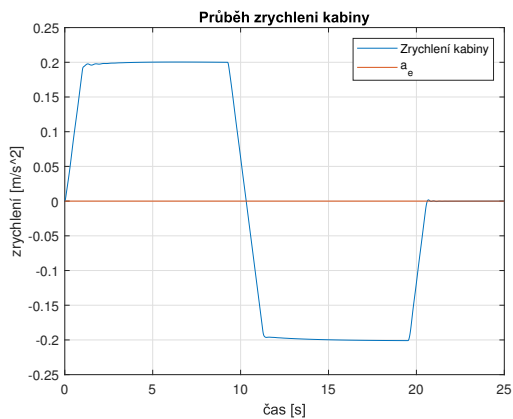


Obrázek 56: Schéma hybridní SDRE stavové - PI robustní regulace

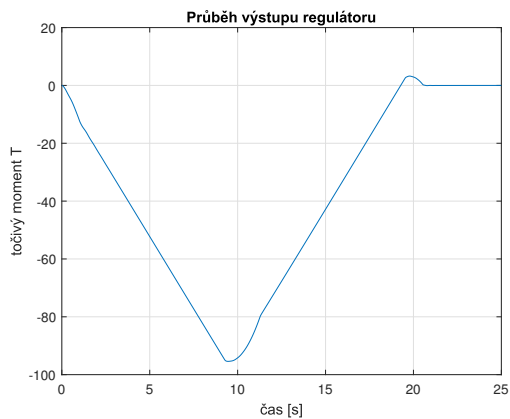
Jak můžeme vidět, schéma dále obsahuje vedle stavové zpětné vazby i náš robustní regulátor (na rozdíl od robustní regulace zde ale již není notch filtr). Ten by měl stavové zpětné vazbě významně vypomoci, jelikož bude udávat točivý moment podle generátoru trajektorie, a SZV bude již jen korigovat odchylky od této trajektorie (podle které byl linearizován systém, pro který je tato časově variantní SZV navrhována). To můžeme vidět na výsledcích simulace jízdy dolů - [obrázek 57](#). Výsledky simulace jízdy vzhůru jsou rovněž významně lepší - [obrázek 58](#).

Můžeme vidět, že tato hybridní regulace dosahuje ještě menších odchylek (téměř polovičních oproti samotnému LQR), referenci rychlosti tedy sleduje s maximální odchylkou $17\text{mm} \cdot \text{s}^{-1}$. Významně potlačena je rovněž odchylka polohy v čase dojezdu ($t = 20.5\text{s}$), kterou pak musí výtah dojet pomalu. Celková doba jízdy výtahu je ovšem stejná, hybridní regulace je však významně přesnější.

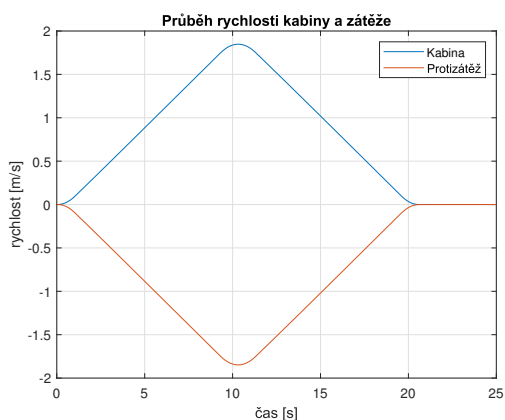
Je nutno znovu upozornit, že tyto výsledky jsme dostali při znalosti celého vektoru stavu. Ten ale k dispozici nemáme, je potřeba jej získat rekonstruktorem. To tedy parametry regulace pravděpodobně o něco zhorší. Po zavedení rekonstrukturu proto matici Q ještě iterativně vylepšíme, abychom získali zcela funkční dynamický kompenzátor.



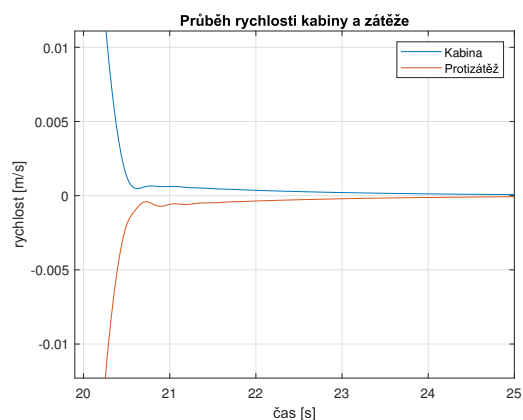
(a) Zrychlení kabiny



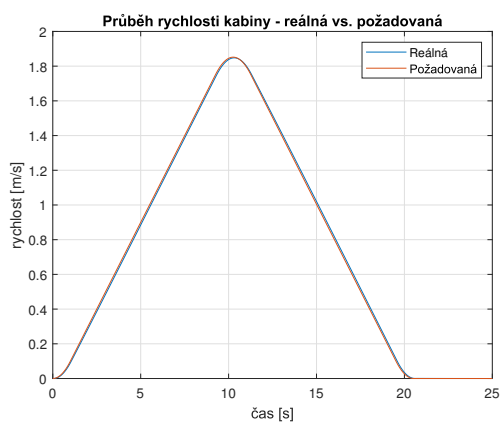
(b) Výstup regulátoru



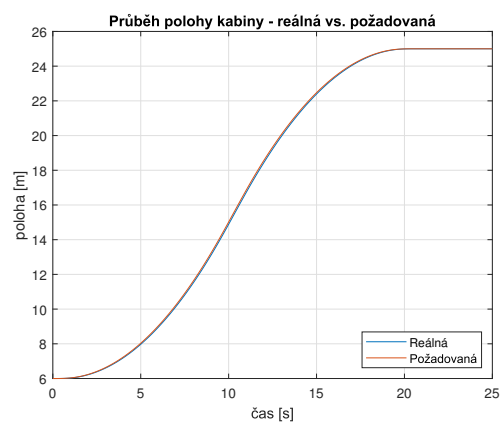
(c) Rychlost kabiny a protizátěže



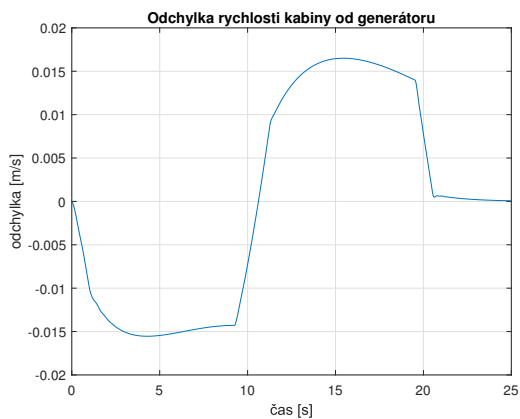
(d) Rychlost kabiny a protizátěže - detail



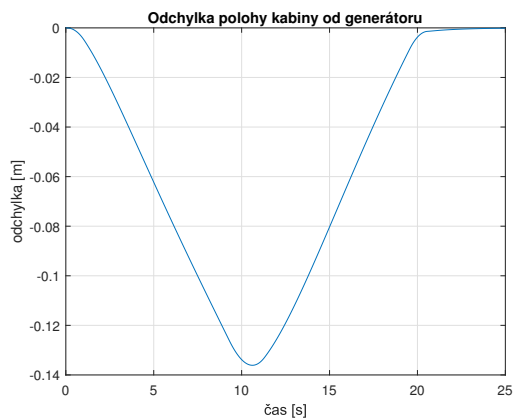
(e) Rychlost kabiny pož. vs. reálná



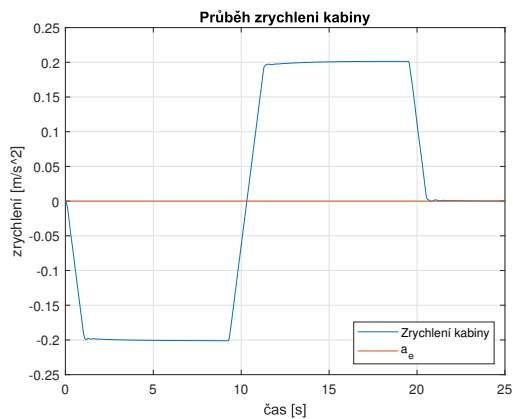
(f) Poloha kabiny pož. vs. reálná



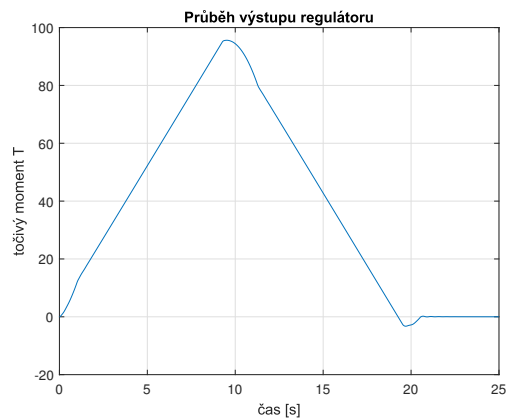
(g) Odchylka rychlosti kabiny od pož.



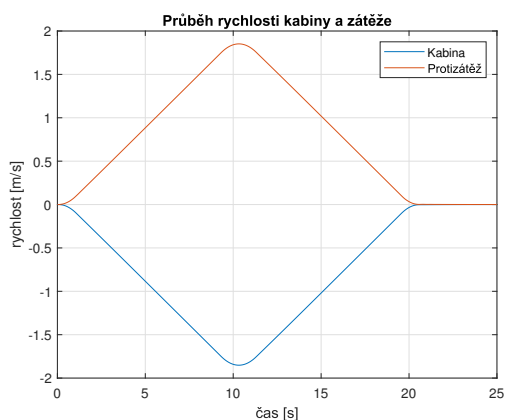
(h) Odchylka polohy kabiny od pož.



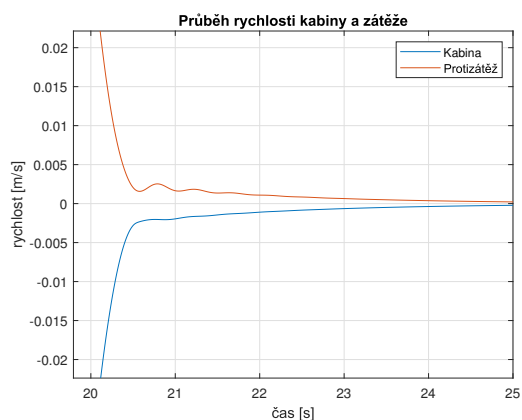
(a) Zrychlení kabiny



(b) Výstup regulátoru



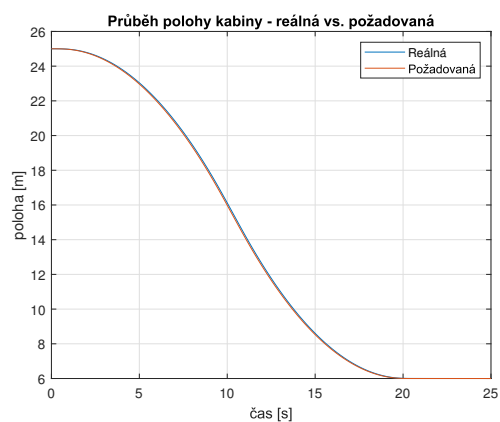
(c) Rychlost kabiny a protizátěže



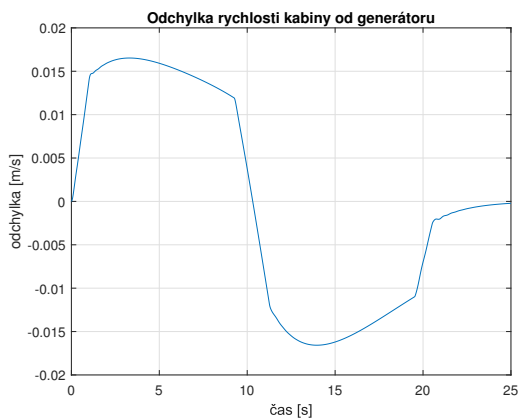
(d) Rychlost kabiny a protizátěže - detail



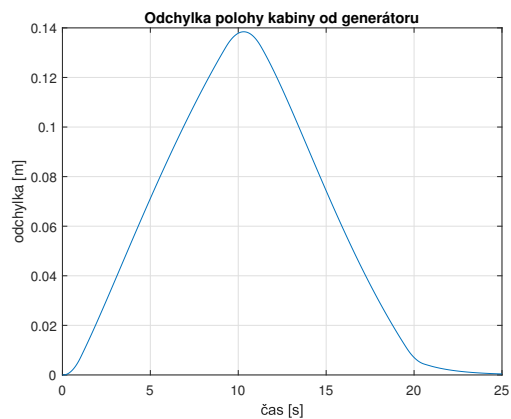
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná



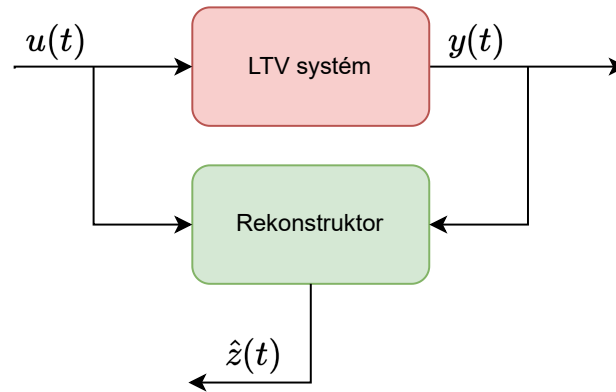
(g) Odchyłka rychlosti kabiny od pož.



(h) Odchyłka polohy kabiny od pož.

6 Rekonstruktor

Jak jsme již řekli, stav, který potřebujeme znát pro stavovou zpětnou vazbu, k dispozici nemáme. Jeho odhad však lze získat s pomocí tzv. rekonstruktoru stavu. To je systém, který obsahuje co nejpřesnější odhad vnitřní dynamiky řízeného systému a na základě vstupu do a výstupu z něj zvládne rekonstruovat jeho vnitřní stav (respektive dá nám jeho odhad $\hat{z}(t)$).



Obrázek 59: Princip fungování rekonstruktoru

Vyjdeme z rovnice rekonstruktoru^[8]

$$\dot{\hat{z}}(t) = \mathbf{A}\hat{z}(t) + \mathbf{B}u(t) + \mathbf{L}(y(t) - \hat{y}(t)) \quad (44)$$

$$\Rightarrow \dot{\hat{z}}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\hat{z}(t) + \mathbf{B}u(t) + \mathbf{L}y(t) \quad (45)$$

kde $(\mathbf{A} - \mathbf{L}\mathbf{C})$ je matice dynamiky chyby rekonstrukce:

$$\dot{\varepsilon}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\varepsilon(t) \Rightarrow \varepsilon(t) = e^{(\mathbf{A} - \mathbf{L}\mathbf{C})t}\varepsilon(0) \quad (46)$$

Pro matici $\mathbf{C} \in \mathbb{R}^{n \times m}$ jsou pak rozměry matice $\mathbf{L} \in \mathbb{R}^{m \times n}$. Dynamiku rekonstruktoru poté určuje charakteristický polynom:

$$a_z = \det(z\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C}) \quad (47)$$

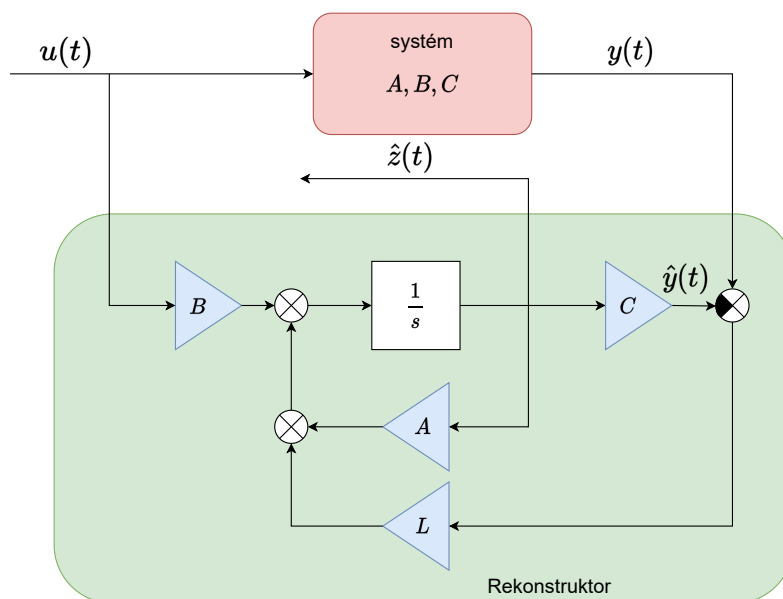
Dynamiku rekonstrukce můžeme přiřadit metodou přiřazení pólů:

$$a_z = \det(z\mathbf{I} - \mathbf{A} + \mathbf{L}\mathbf{C}) \stackrel{!}{=} a_z^* \quad (48)$$

Kde a_z^* je požadovaný polynom. K samotným výpočtům využijeme příkaz `place()`.

Rekonstruktor tedy navrhujeme pro systém linearizovaný v nějaké konkrétní výšce x_{20} . Bylo by možno jej navrhnout i časově proměnný pro systém linearizovaný podél trajektorie, my jsme však došli k závěru, že nám běžný rekonstruktor bude stačit.

^[8]KKY/LS2



Obrázek 60: Vnitřní schéma rekonstruktoru

Nutnou podmínkou k návrhu rekonstruktoru je však pozorovatelnost systému (A, C) . V našem případě se však ukazuje, že systém pozorovatelný není. Rovněž nám vyšlo, že je systém (A, B) neřiditelný, což je už z principu nesmysl, protože jsme systém doteď úspěšně řídili (resp. řídili jsme nelineární systém ze kterého linearizovaný vychází a měl by být snazší na řízení). Podrobně jsme tudíž prozkoumali obecnou matici \mathbf{A} (vztah 18):

$$\frac{\partial f}{\partial \mathbf{z}} = \mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_0}{m_1 l_1} & 0 & \frac{k_0 r z_1}{m_1 l_1^2} & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \frac{-k_0}{m_2 l_2} & \frac{-k_0 r z_2}{m_2 l_2^2} & 0 & \frac{-b_0}{m_2 l_2} & \frac{-b_0 r}{m_2 l_2} \\ \frac{k_0 r}{J l_1} & \frac{-k_0 r}{J l_2} & \frac{-k_0 r^2 (z_2 l_1^2 + z_1 l_2^2)}{J l_1^2 l_2^2} & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2 (l_1 + l_2) b_0 - b_f l_1 l_2}{J l_1 l_2} \end{bmatrix}$$

A zjistili jsme, že sloupce 1,2 a 3 jsou skutečně lineárně závislé, konkrétně $s_3 = \frac{-r z_1}{l_1} s_1 + \frac{r z_2}{l_2} s_2$. Logicky nám tudíž nemůže vyjít hodnota matice \mathbf{A} plná a tak nemůže vyjít plná ani hodnota matice pozorovatelnosti $Q_p = [C; CA; CA^2; \dots; CA^6]$. Lidsky řečeno to znamená, že pozice motoru je vypočitatelná z pozic kabiny a protizátěže a tudíž je v naší stavové reprezentaci redundantní.

Bylo by možno prostě využít příkaz `minreal()` v Matlabu, který by jeden stav eliminoval a vhodně přepočítal všechny matice, z reprezentace by se pak ale vytratil fyzikální charakter systému a nebylo by možno zpětně rekonstruovat absolutní polohy x_1, x_2, ϕ .

To napravíme zavedením nových stavů - torze mezi pozicí protizátěže a pozicí motoru $\delta_1 = x_1 - r\phi + r\pi$ a torze mezi pozicí kabiny a pozicí motoru $\delta_2 = x_2 + r\phi - l_0$.

Je samozřejmě nutno i vhodně přepočítat matici \mathbf{A} - vyloučíme třetí řádek a třetí sloupec, ovšem je nutno vhodně upravit vztahy ve sloupcích a řádcích 1 a 2. Pozice $\mathbf{A}(4 : 6, 4 : 6)$ zůstanou nezměněny:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 & -r \\ 0 & 0 & 0 & 1 & +r \\ \xi_1 & 0 & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \xi_2 & 0 & \frac{-b_0}{m_2 l_2} & \frac{b_0 r}{m_2 l_2} \\ \psi_1 & \psi_2 & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2(l_1+l_2)b_0 - b_f l_1 l_2}{J l_1 l_2} \end{bmatrix} \quad (49)$$

Prvky ξ_i, ψ_i musíme odvodit opět od nuly z rovnic 6, 7, 8:

$$\begin{aligned} \ddot{x}_1 &= g - \frac{k_0}{m_1(\phi r - \pi r)}(x_1 - \phi r + \pi r) - \frac{b_0}{m_1(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) - \frac{F_{f1}}{m_1} \\ \ddot{x}_2 &= g - \frac{k_0}{m_2(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0}{m_2(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - \frac{F_{f2}}{m_2} \\ \ddot{\phi} &= \frac{T}{J} + \frac{k_0 r}{J(\phi r - \pi r)}(x_1 - \phi r + \pi r) + \frac{b_0 r}{J(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) + \dots \\ &\dots - \frac{k_0 r}{J(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0 r}{J(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - b_f \frac{\dot{\phi}}{J} \end{aligned}$$

Které prepíšeme do tvaru s uvažovanými novými stavy - zkruty δ_1, δ_2 :

$$\ddot{x}_1 = g - \frac{k_0}{m_1(x_1 - \delta_1)}(\delta_1) - \frac{b_0}{m_1(x_1 - \delta_1)}(\dot{\delta}_1) - \frac{F_{f1}}{m_1} \quad (50)$$

$$\ddot{x}_2 = g - \frac{k_0}{m_2(x_2 - \delta_2)}(\delta_2) - \frac{b_0}{m_2(x_2 - \delta_2)}(\dot{\delta}_2) - \frac{F_{f2}}{m_2} \quad (51)$$

$$\begin{aligned} \ddot{\phi} &= \frac{T}{J} + \frac{k_0 r}{J(x_1 - \delta_1)}(\delta_1) + \frac{b_0 r}{J(x_1 - \delta_1)}(\dot{\delta}_1) + \dots \\ &\dots - \frac{k_0 r}{J(x_2 - \delta_2)}(\delta_2) - \frac{b_0 r}{J(x_2 - \delta_2)}(\dot{\delta}_2) - b_f \frac{\dot{\phi}}{J} \end{aligned} \quad (52)$$

$$\delta_1 = x_1 - r\phi + r\pi, \quad \delta_2 = x_2 + r\phi - l_0, \quad \dot{\delta}_1 = \dot{x}_1 - r\dot{\phi}, \quad \dot{\delta}_2 = \dot{x}_2 + r\dot{\phi} \quad (53)$$

Z nich můžeme odvodit parciálními derivacemi prvky ξ_i, ψ_i :

$$\begin{aligned} \xi_1 &= \frac{\partial \ddot{x}_1}{\partial \delta_1} = \frac{-k_0 x_1}{m_1 l_1^2} \\ \xi_2 &= \frac{\partial \ddot{x}_2}{\partial \delta_2} = \frac{-k_0 x_2}{m_2 l_2^2} \\ \psi_1 &= \frac{\partial \ddot{\phi}}{\partial \delta_1} = \frac{k_0 r x_1}{J l_1^2} \\ \psi_2 &= \frac{\partial \ddot{\phi}}{\partial \delta_2} = \frac{-k_0 r x_2}{J l_2^2} \end{aligned} \quad (54)$$

Nový stavový popis bude tedy vypadat následovně:

$$\begin{aligned}
 \zeta &= [\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5] = [\delta_1, \delta_2, \dot{x}_1, \dot{x}_2, \dot{\phi}] \\
 \mathbf{A} &= \begin{bmatrix} 0 & 0 & 1 & 0 & -r \\ 0 & 0 & 0 & 1 & +r \\ \frac{-k_0 x_1}{m_1 l_1^2} & 0 & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \frac{-k_0 x_2}{m_2 l_2^2} & 0 & \frac{-b_0}{m_2 l_2} & \frac{-b_0 r}{m_2 l_2} \\ \frac{k_0 r x_1}{J l_1^2} & \frac{-k_0 r x_2}{J l_2^2} & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2 (l_1 + l_2) b_0 - b_f l_1 l_2}{J l_1 l_2} \end{bmatrix} \\
 \mathbf{B} &= [0 \ 0 \ 0 \ 0 \ \frac{1}{J}]^T ; \mathbf{C} = [0 \ 0 \ 0 \ 0 \ 1] ; \mathbf{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 \dot{\zeta}(t) &= \mathbf{A}\zeta(t) + \mathbf{B}u(t) \\
 \mathbf{y}(t) &= \mathbf{C}\zeta(t) + \mathbf{D}u(t)
 \end{aligned} \tag{55}$$

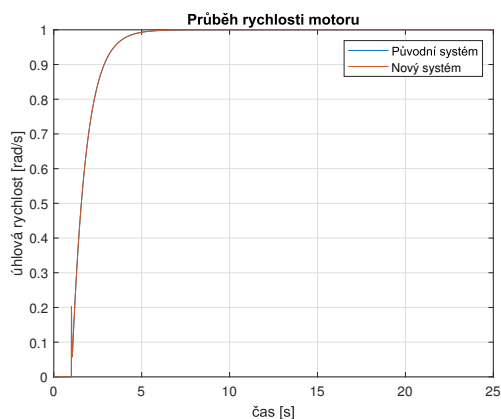
Ve stavu tedy vůbec nefiguruje poloha motoru ϕ a navíc se ukazuje, že jeho absolutní poloha nemá vůbec žádný vliv na zrychlení kabiny, protizátěže nebo dokonce sebe sama. Zdá se to nelogické, avšak jen do chvíle než si uvědomíme, že lano je modelováno jako pružina, tz. zrychlení jejích dvou konců (kabiny a motoru) skutečně nezáleží na tom, v jaké absolutní výšce se reálně nachází, ale pouze na tom, jak je stlačená (roztažená), což v našem případě právě říká stav torze δ . Samozřejmě, vzhledem k naší nelinearitě je pružnost lana (tz. pomyslné pružiny) závislá na jeho délce, proto v parciálních derivacích vystupují výšky x_1 a x_2 , za které se dosadí konkrétní linearizační výšky (z x_2 vypočteme x_1), a také délky l_1, l_2 , čímž se v podstatě vypočítá pružnost lana příslušná konkrétnímu RB ve kterém linearizujeme.

Tato reprezentace by měla být vstupně-výstupně ekvivalentní^[9] s původním systémem, ovšem ze simulací se ukazuje, že to není tak docela pravda.

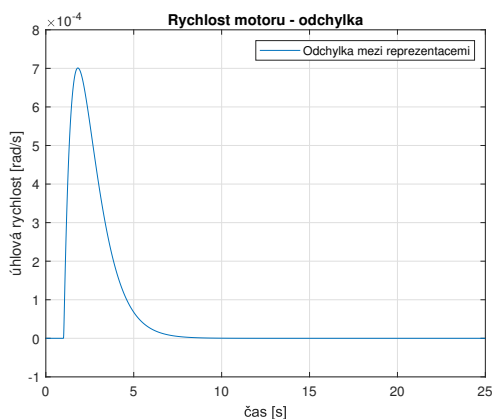
Nejprve uvedeme odezvu systémů na jednotkový skok - [obrázek 61](#). Odchylka $10^{-4} \text{ rad} \cdot \text{s}^{-1}$ se může zdát malá, u IOE systémů by ale měla být na úrovni numerické chyby (např. 10^{-10}).

Vše jsme tedy několikrát znovu přepočítali, provedli znovu kompletně celou linearizaci a prověřili dosazení hodnot v Matlabu, i tak jsme ale došli k naprosto stejným výsledkům.

^[9]Jako IOE uvažujeme systémy, které na stejný vstup reagují stejným výstupem. Vnitřní reprezentace nás v tomto případě nezajímá, nejedná se o transformaci systému maticí, ale o zmenšení reprezentace o jeden (redundantní) stav.



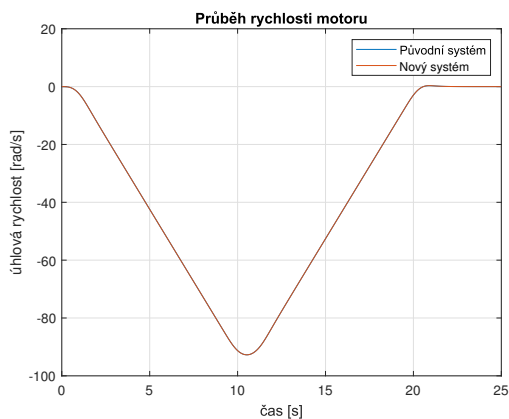
(a) Rychlost motoru



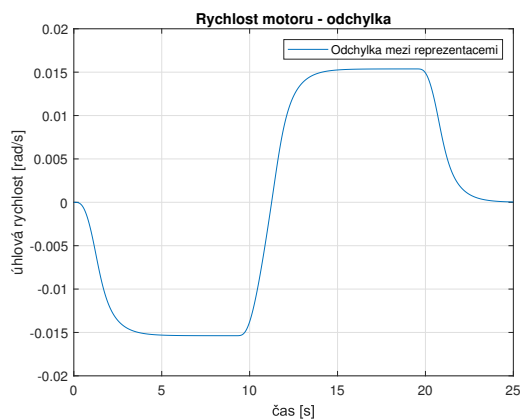
(b) Odchylka rychlostí motoru

Obrázek 61: Test ekvivalence systémů

Když uvažíme odezvu systémů na generátor požadované trajektorie (s robustním regulátorem), můžeme vidět (viz [obrázek 62](#)), že charakter odchylky rychlosti motoru těchto systémů se velmi podobá odchylce rychlosti kabiny od generátoru (např. [obrázek 58g](#)).



(a) Rychlost motoru



(b) Odchylka rychlostí motoru

Obrázek 62: Test ekvivalence systémů

Začali jsme tedy detailně zkoumat, čím by to mohlo být způsobeno, jelikož tato reakce naznačuje, že původní systém reaguje na vstupy opožděně oproti novému (torznímu). To je nesmysl, protože žádný ze systémů neobsahuje žádné dopravní zpoždění.

Po použití příkazu *minreal()* na původní systém a porovnání pólů a nul tohoto a nového (torzního) systému (oba linearizovány pro $x_2 = 16m, m_2 = 1000kg$) jsme zjistili, že nový systém má stejné reálné části nul, imaginární části jsou ale kmitavější (v řádech setin), u pólů se pak liší i reálné části, viz [tabulka 1](#):

	Původní systém	Nový systém
Póly	-312.5123 + 0.0000i -77.7519 + 0.0000i -1.8222 + 17.3287i -1.8222 - 17.3287i -1.2383 + 0.0000i	-312.5106 + 0.0000i -77.7512 + 0.0000i -1.8222 + 17.3455i -1.8222 - 17.3455i -1.2407 + 0.0000i
Nuly	-1.7680 +17.0748i -1.7680 -17.0748i -1.8786 +17.5946i -1.8786 -17.5946i	-1.7680 +17.0914i -1.7680 -17.0914i -1.8786 +17.6117i -1.8786 -17.6117i

Tabulka 1: Póly a nuly původního a torzního systému

To tedy znamená, že nový systém je kmitavější a proto reaguje na vstup 'hyperaktivněji' než původní, což způsobuje odchylku. Jelikož jsme nikde nenašli žádnou chybu ve výpočtech ani v dosazení, a protože potřebujeme v práci i přesto pokračovat, uvažujeme následovně - maximální odchylka rychlosti motoru $0.015 \text{ rad} \cdot \text{s}^{-1}$ znamená odchylku rychlosti kabiny do $0.0003m \cdot \text{s}^{-1}$ (protože $r = 0.02$ je převodní konstanta mezi rotačním a lineárním pohybem). Tato odchylka se tedy v případě rekonstruování přičte k odchylce, kterou bude mít systém od generátoru trajektorie, a protože ta je $50 \times$ větší ($0.015m \cdot \text{s}^{-1}$), nebude mít odchylka mezi reprezentacemi systému valný význam.

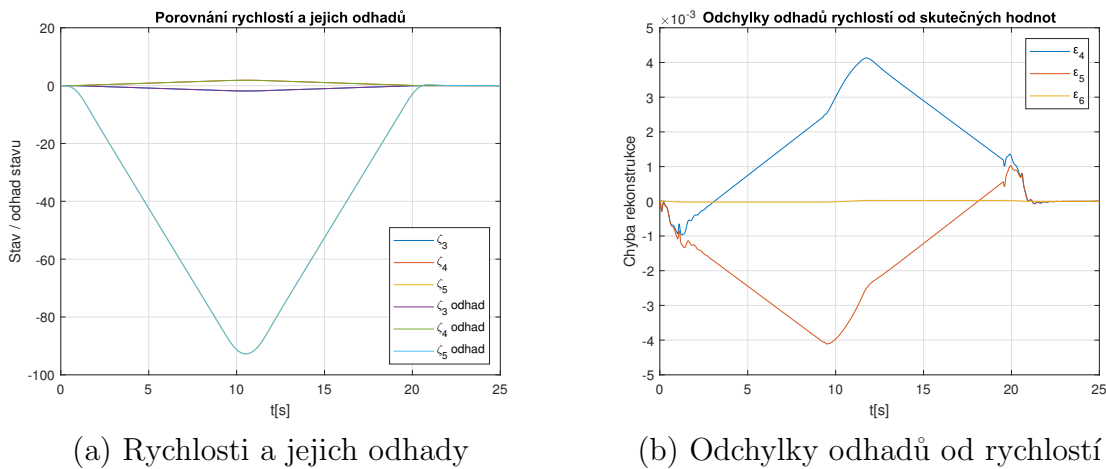
Z těchto důvodů tedy budeme systémy dále považovat za IOE (byť striktně vzato nejsou).

Redukovaný systém má tedy pouze 5 pólů a tudíž i jemu příslušný rekonstruktor jich bude mít stejný počet, jeho matice \mathbf{L} pak bude rozměru 5×1 .

Protože póly systému budou rychlejší pro menší x_2 , zjistíme póly systému pro $x_2 = 6m$, kde budou v rámci naší trajektorie nejrychlejší. To děláme proto, že póly rekonstruktoru musí být rychlejší než systému, jinak by rekonstrukce nefungovala.

Protože póly tohoto systému jsou $P_S = [-510.474, -80.395, -2.062 + 17.267i, -2.062 - 17.267i, -1.241]$ (pro $x_2 = 6m, m_2 = 1000kg$), zvolíme póly rekonstruktoru o něco rychlejší (konkrétní hodnoty volíme metodou pokus-omyl): $P_R = [-700, -400, -25 + 20i, -25 - 20i, -20]$.

Systém tedy nyní linearizujeme ve výšce $x_2 = 16m$ a pro něj vytvoříme rekonstruktor s danými póly P_R . Pro simulaci pohybu podél naší trajektorie (nelineární systém s robustním regulátorem) pak vypadá rekonstrukce rychlostí $\dot{x}_1, \dot{x}_2, \dot{\phi}$ následovně (obrázek 63):



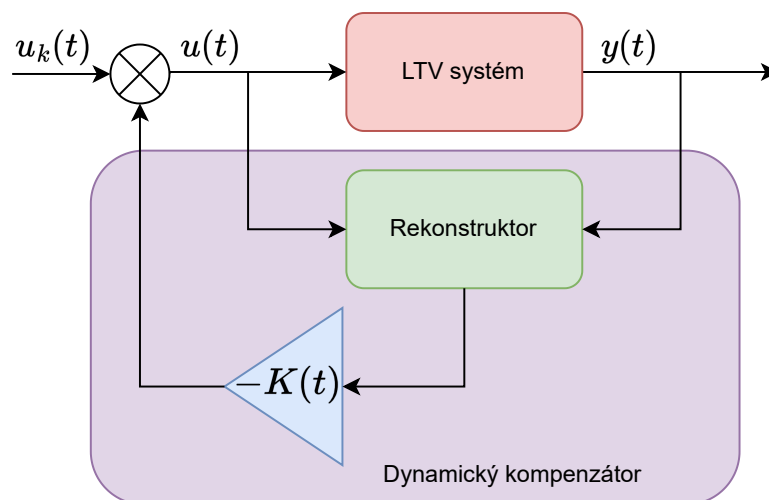
Obrázek 63: Odhady rychlostí systému a odchyly od skutečných pro simulaci podél trajektorie

Jak jsme již řekli, záleží nejen na zvolených pólech, ale i na tom, v jaké výšce x_2 je systém linearizován. Linearizovat systém uprostřed mezi výškou počáteční a konečnou se může jevit jako vhodné, ovšem není tomu tak. Můžeme si všimnout, že odchyly odhadů rychlostí \dot{x}_1, \dot{x}_2 na začátku a na konci poněkud oscilují. To bude v případě stavové zpětné vazby působit problémy (více v [sekcí 7](#)) a proto nebudeme systém linearizovat ve výšce $x_2 = 16m$, ale v počáteční výšce $x_2 = x_{20} = 6m$, aby rekonstruktor pokud možno co nejlépe sledoval rychlosti na začátku pohybu (rekonstruktor se časem naladí na systém pomocí inovativní vazby \mathbf{L} a pak již bude rekonstruovat dostatečně dobře).

Dále bude též problém s tím, že rekonstruktor neumí rekonstruovat polohy x_1, x_2, ϕ ale pouze rozdíl v jejich vzájemné poloze (torze δ_1, δ_2). O tom ale v další kapitole.

7 SDRE dynamický kompenzátor

Jak jsme již řekli, dynamický kompenzátor je vlastně jen souhrnný pojem pro spojení stavové zpětné vazby (bez ohledu na způsob, jakým jsme jí získali - zda přiřazením pólů nebo LQ úlohou) a rekonstruktoru. V praxi totiž málokdy známe všechny stavy a ty neznámé je tudíž potřeba nějakým způsobem odhadnout. Po získání všech stavů (ať už změřených nebo rekonstruovaných) již můžeme použít stavovou zpětnou vazbu k řízení systému.



Obrázek 64: Schéma dynamického kompenzátoru

Nyní nastává problém rekonstrukce poloh x_1, x_2, ϕ . Rekonstrukce polohy motoru ϕ je jednoduchá, stačí integrovat rychlost $\dot{\phi}$, kterou sledujeme přímo (není potřeba jí rekonstruovat) a tudíž takto získanou polohu můžeme považovat za velmi přesnou.

Dále je potřeba získat polohy protizátěže a kabiny. Šlo by tak učinit ze znalosti zrekonstruovaného ϕ a z rekonstruovaných δ_1, δ_2 . Avšak ukázalo se, že tento přístup není úplně vhodný, protože mezi stavy $\dot{x}_1, \dot{x}_2, \dot{\phi}$ a stavy δ_1, δ_2 je ve své podstatě vztah integrace, a tudíž pokud se rekonstruované rychlosti \dot{x}_1, \dot{x}_2 byť jen trochu liší od těch skutečných, naintegruje se tato chyba do stavů δ_1, δ_2 , a tudíž když je využijeme k rekonstrukci poloh, budou tyto polohy posunuty o integrační chybu.

Ty se ukázaly být tak velké, že se stala stavová regulace nepoužitelnou. Proto jsme se rozhodli být inovativní a vymysleli způsob, jak tuto chybu významně potlačit při zachování schopnosti vysledovat poruchy (jak je možno vidět později v [sekci 8.2](#) na [obrázku 83](#)).

Místo využití torzí získaných z rekonstruktoru vypočítáme, jaká by torze v daném místě trajektorie **měla být**, což můžeme díky tomu, že pro všechny stavy $x_1, x_2, \phi, \dot{x}_1, \dot{x}_2, \dot{\phi}$ máme vytvořený generátor trajektorie. Tyto torze budeme značit jako δ_{1p}, δ_{2p} (pracovní) a platí pro ně vztahy:

$$\delta_{1p} = x_{1p} - r\phi_p + r\pi, \quad \delta_{2p} = x_{2p} + r\phi_p - l_0 \quad (56)$$

Nyní využijeme definice δ_1 a δ_2 k odvození \hat{x}_1, \hat{x}_2 :

$$x_1 = \delta_1 + r\phi - r\pi \Rightarrow \hat{x}_1 = \delta_{1p} + r\hat{\phi} - r\pi \quad (57)$$

$$x_2 = \delta_2 - r\phi + l_0 \Rightarrow \hat{x}_2 = \delta_{2p} - r\hat{\phi} + l_0 \quad (58)$$

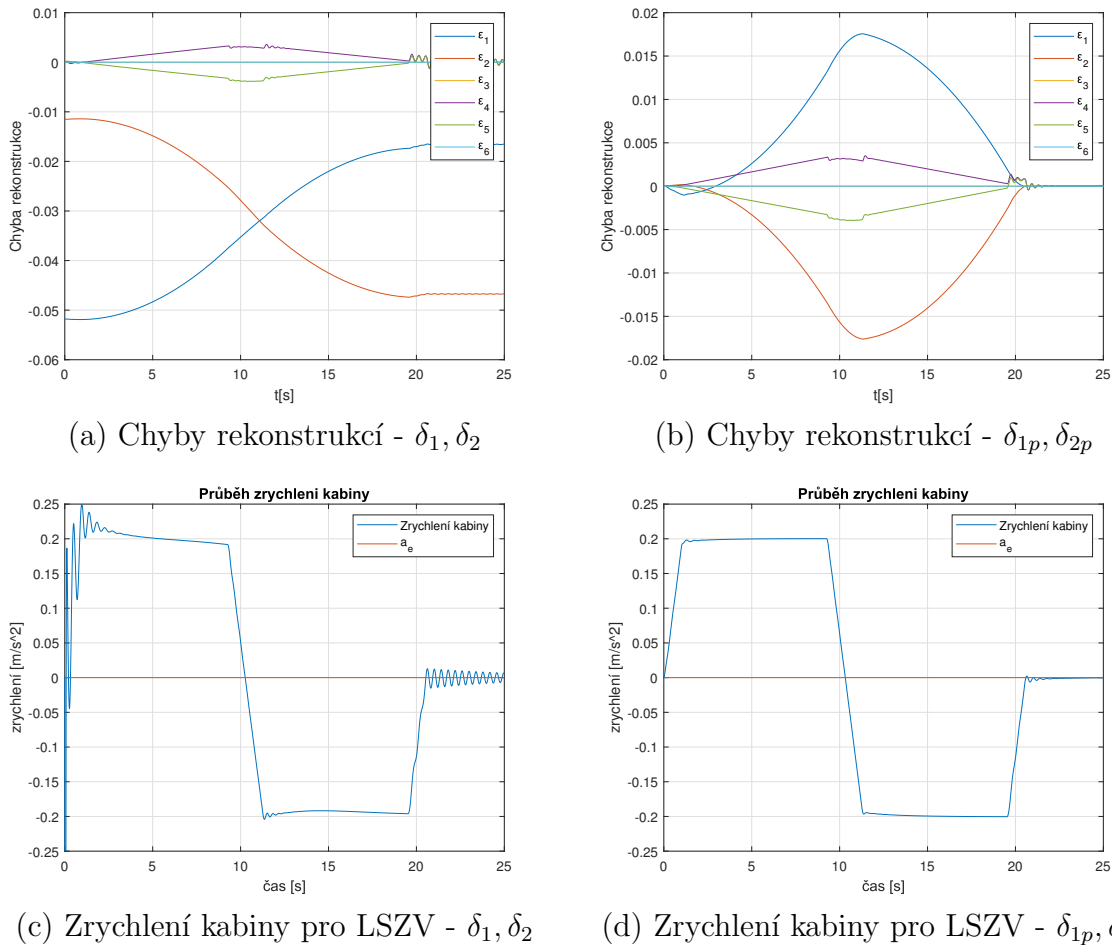
Po zkombinování tedy získáme:

$$\hat{x}_1 = x_{1p} - r\phi_p + r\pi + r\hat{\phi} - r\pi = x_{1p} + r(\hat{\phi} - \phi_p) \quad (59)$$

$$\hat{x}_2 = x_{2p} + r\phi_p - l_0 - r\hat{\phi} + l_0 = x_{2p} - r(\hat{\phi} - \phi_p) \quad (60)$$

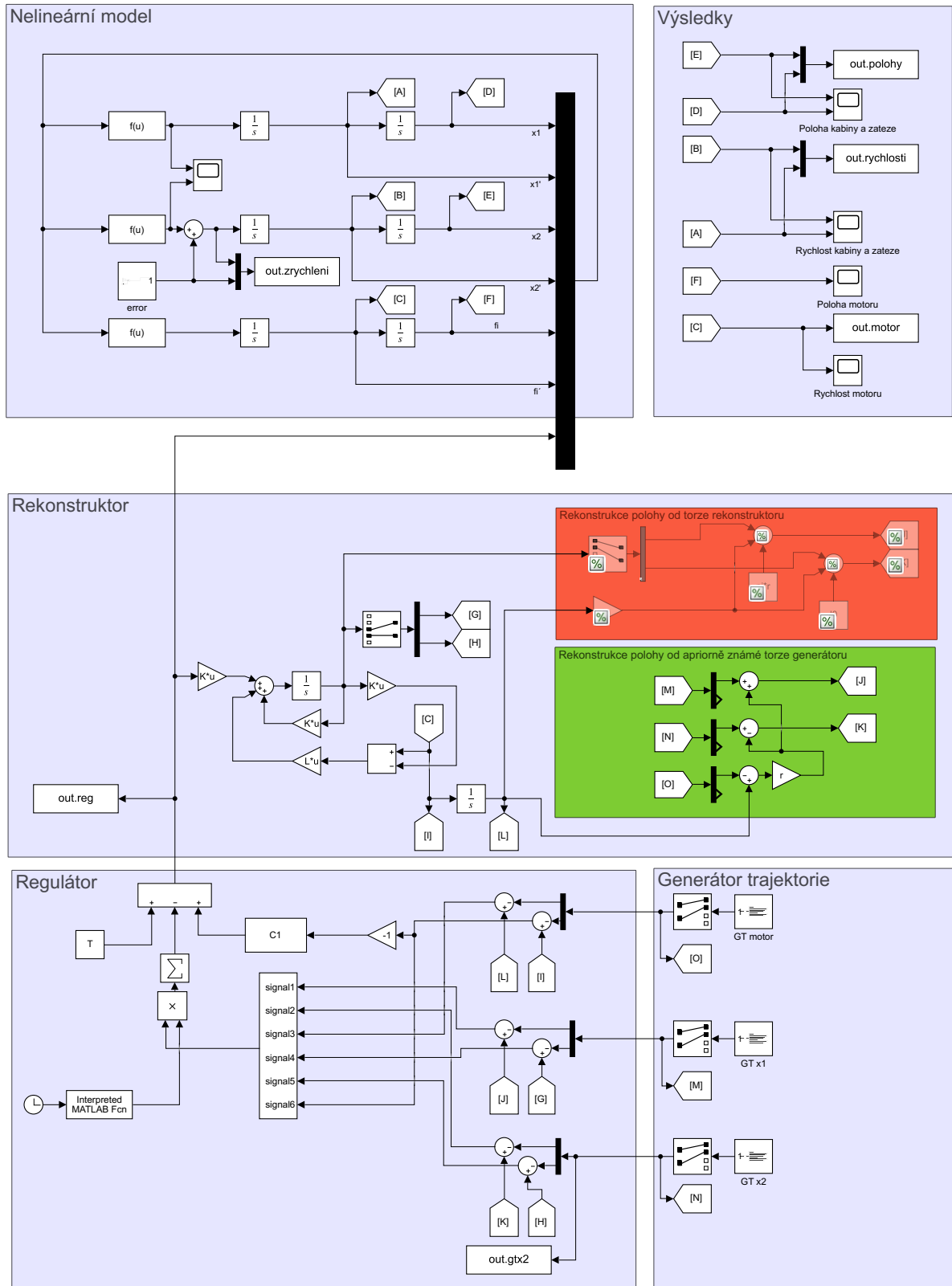
To tedy znamená, že pro úspěšnou rekonstrukci poloh x_1, x_2 poblíž pracovní trajektorie jsou potřeba pouze jejich požadované hodnoty x_{1p}, x_{2p} v daný moment a poté rozdíl mezi pracovním ϕ_p a jeho reálnou (odhadovanou) polohou $\hat{\phi}$. Právě využití velmi přesného odhadu $\hat{\phi}$ zajišťuje, že umí rekonstruktor zachytit i poruchy.

O tom, že tento přístup značně vylepší rekonstrukci, svědčí simulace (obrázek 65):



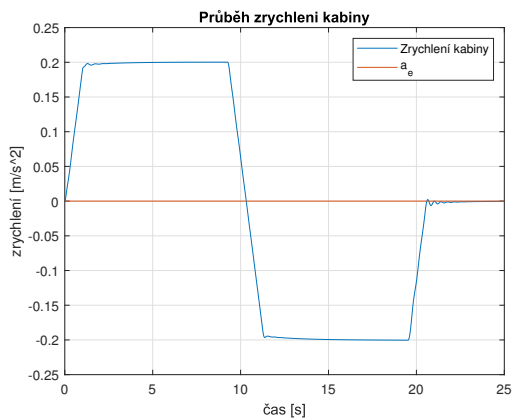
Obrázek 65: Rozdíly ve kvalitě rekonstrukce a z ní vyplývající LSZV

Schéma dynamického kompenzátoru s vyznačenými rozdílnými druhy rekonstrukce - **červeně** s použitím rekonstruovaných δ_1, δ_2 a **zeleně** s použitím apriorně známých δ_{1p}, δ_{2p} - obrázek 66:

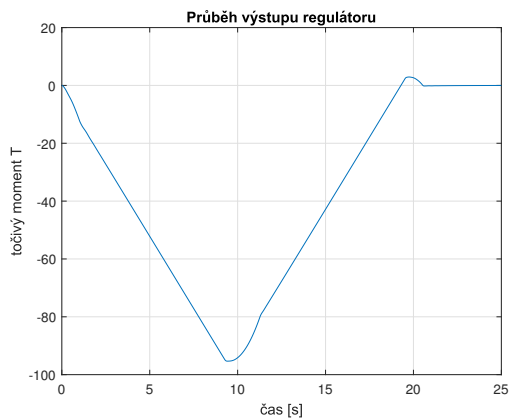


Obrázek 66: Implementace dynamického kompenzátoru v Simulinku

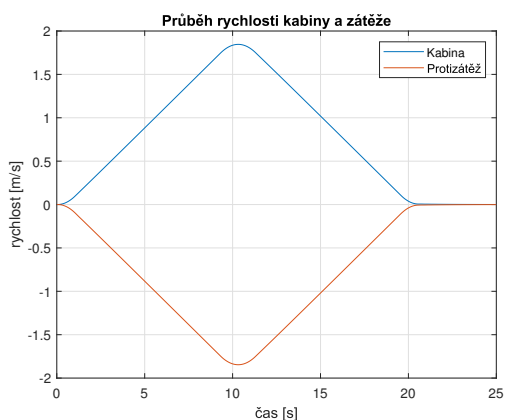
Výsledky simulace řízení s využitím hybridní LQR stavové - PI robustní regulace, rekonstruktoru a apriorně známých torží δ_{1p}, δ_{2p} - obrázek 67:



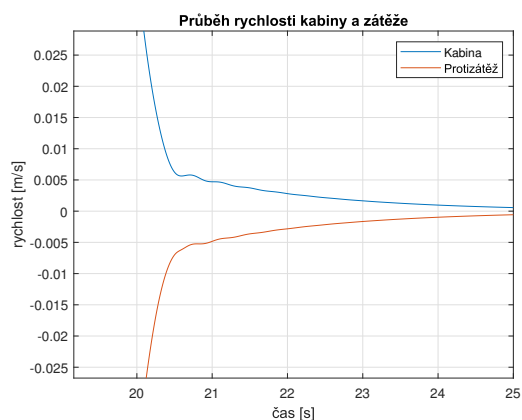
(a) Zrychlení kabiny



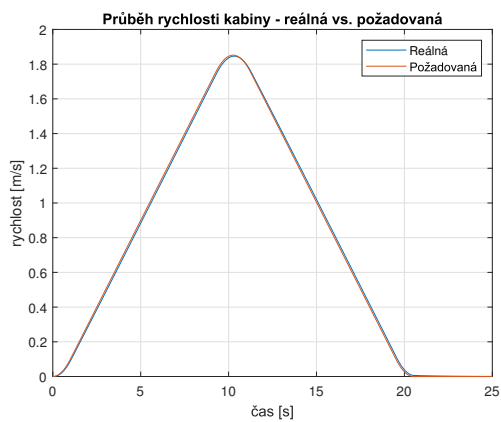
(b) Výstup regulátoru



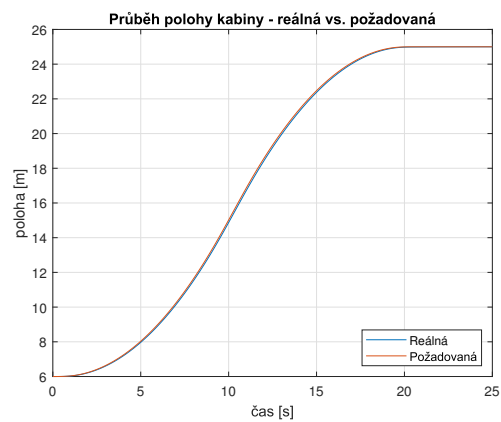
(c) Rychlost kabiny a protizátěže



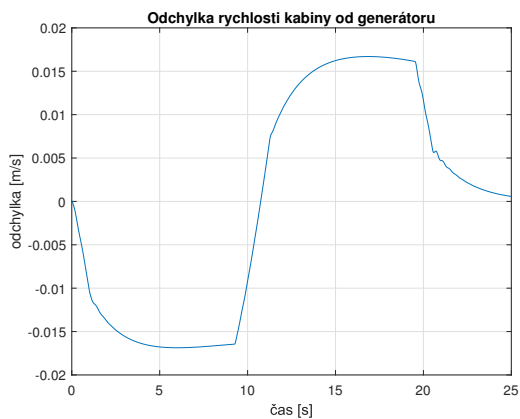
(d) Rychlost kabiny a protizátěže - detail



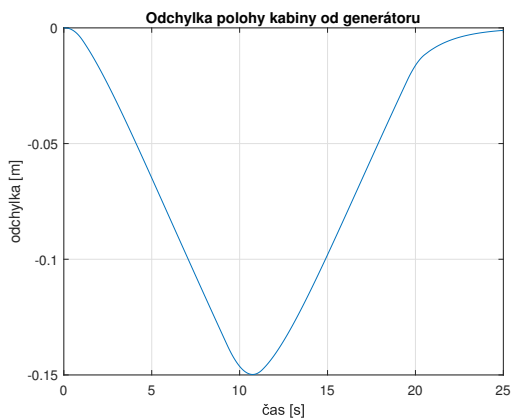
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná



(g) Odchylka rychlosti kabiny od pož.



(h) Odchylka polohy kabiny od pož.

8 Simulace s působením poruch

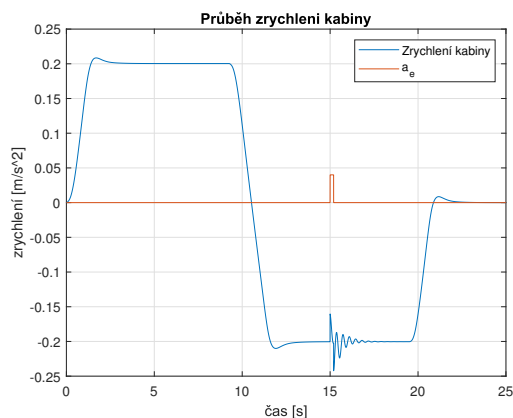
8.1 Porucha na kabině a_e

Simulace jsme doteď prováděli bez působení poruch, čímž jsme zkoumali výhradně řízení typu servo (tz. sledování požadované trajektorie). V praxi je ovšem velmi důležitá i druhá vlastnost regulátorů, tedy problém odregulování poruch (vstupních, výstupních i jiných).

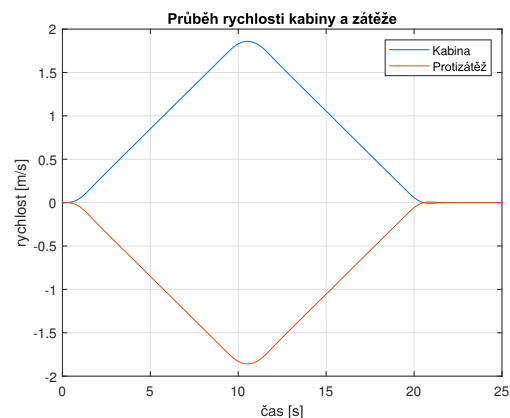
Začneme s robustním regulátorem.

Do simulačního schématu **nelineárního systému** zavedeme poruchu a_e působící přímo na zrychlení kabiny (vnitřní porucha). Tou bude obdélníkový puls o amplitudě $0.04ms^{-2}$ (tz. 20% maximálního zrychlení kabiny) a délce trvání 1s, který může reprezentovat například pohyb lidí v kabině („poskočení“) nebo silové rázy od nerovného lineárního vedení kabiny. Vzhledem k hmotnosti kabiny to odpovídá síle $40N$.

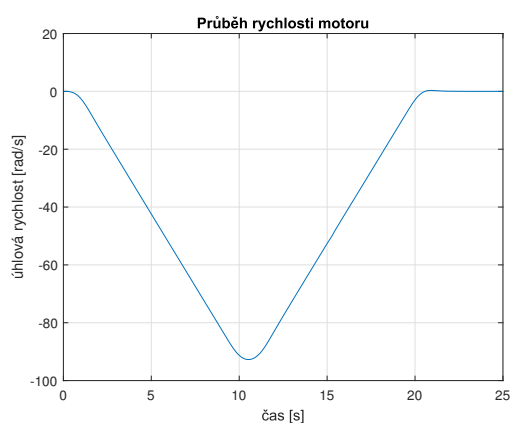
Jak můžeme vidět na [obrázku 68](#), chyba této velikosti se ze zrychlení kabiny téměř vůbec 'nepropíše' na její rychlost (vzniklá odchylka zrychlení není dostatečně velká na to, aby významněji pozměnila průběh rychlosti kabiny), tudíž ani na rychlost motoru přes kterou regulujeme, a tak na ní regulátor zareaguje takřka nezatelně.



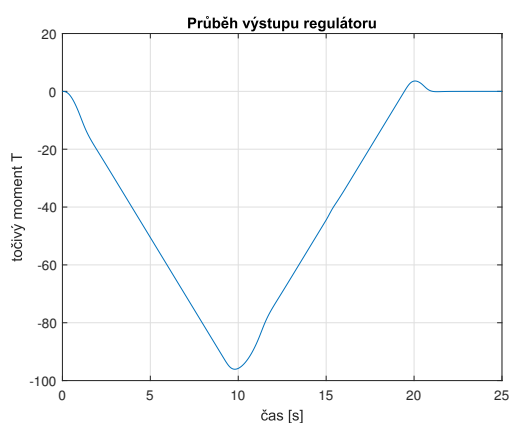
(a) Zrychlení kabiny



(b) Rychlost kabiny a protizátěže



(c) Rychlost motoru



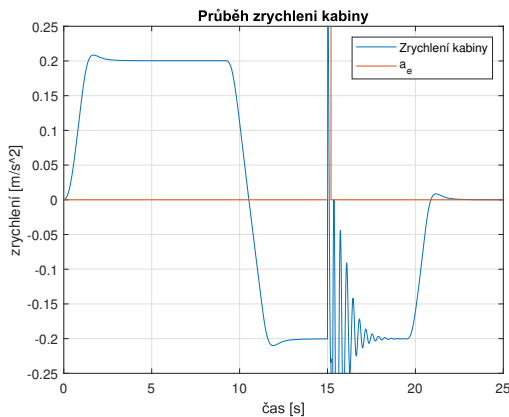
(d) Výstup regulátoru

Obrázek 68: Simulace robustní regulace s poruchou $0.04ms^{-2}$ působící v čase $t = 15s$

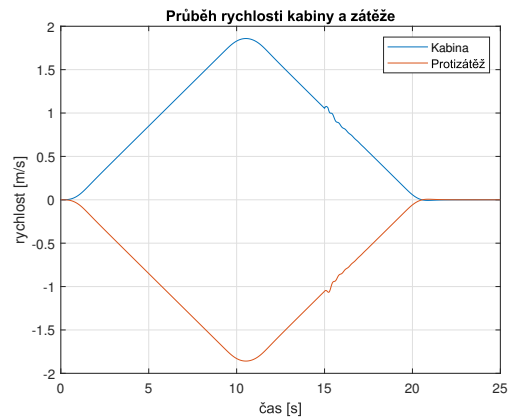
Nyní tedy velikost chyby zvětšíme na 0.6ms^{-2} . Na [obrázku 69](#) můžeme vidět, že chyba této velikosti (odpovídající síle 600N) se již propíše do rychlosti kabiny viditelně, tudíž i do rychlosti motoru, kterou regulátor dorovná. Bez ohledu na velikost poruchy je potlačena za zhruba stejnou dobu (cca 3s).

Z toho tedy vyplývá, že velikost zásahu regulátoru je úměrná její velikosti. Vidíme, že regulátor správně reguluje rychlost kabiny tak, aby odpovídala požadované trajektorii. V případě zrychlení je ten problém, že jeho kmitání nemá regulátor jak zachytit, protože nemáme žádný senzor zrychlení kabiny (ani virtuální - rekonstruktor). Pokud by byl k dispozici, bylo by možno docílit ještě významně lepší regulace.

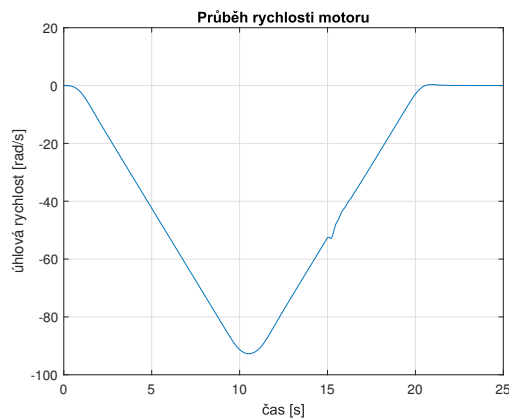
Dále uvedeme ještě [obrázek 70](#) chyby působící v čase dojezdu do správné polohy. Jak vidíme, regulátor i v tomto případě rychlost dorovná.



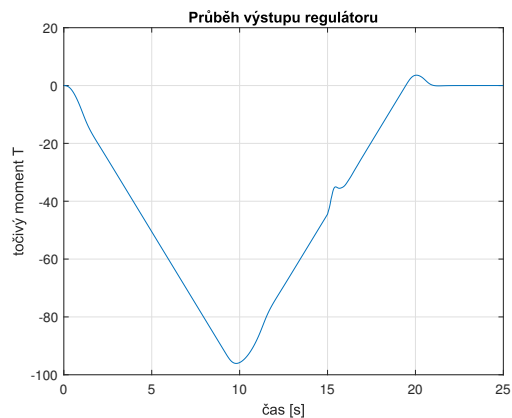
(a) Zrychlení kabiny



(b) Rychlost kabiny a protizátěže

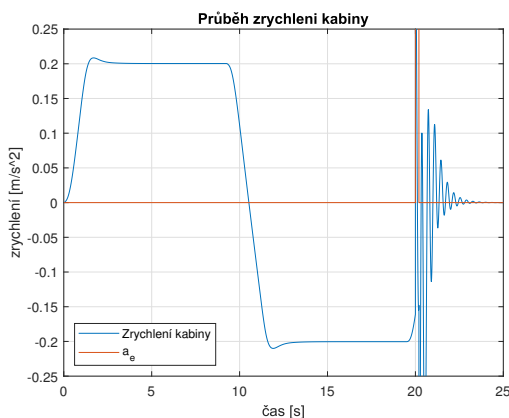


(c) Rychlost motoru

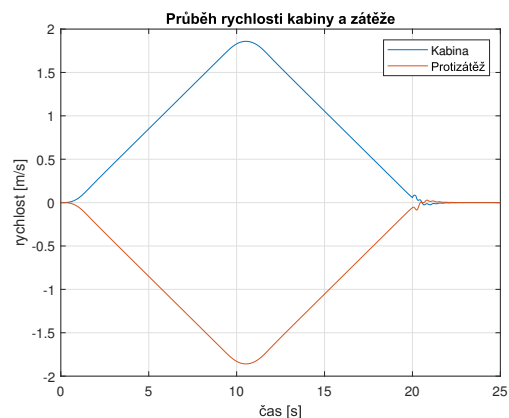


(d) Výstup regulátoru

Obrázek 69: Simulace robustní regulace s poruchou 0.6ms^{-2} působící v čase $t = 15\text{s}$



(a) Zrychlení kabiny



(b) Rychlost kabiny a protizátěže

Obrázek 70: Simulace robustní regulace s poruchou 0.6ms^{-2} působící v čase $t = 20\text{s}$

To, že regulátor koriguje odchylku rychlosti motoru (a tedy i kabiny) vždy stejnou dobu, si můžeme ověřit z přenosu ze zrychlení kabiny na rychlost motoru. Ten získáme tak, že vytvoříme nový systém, který bude mít vstup nejen u motoru (moment T), ale i u kabiny (chybové zrychlení a_e), tedy $\mathbf{u} = [T, a_e]^T$. Dostaneme tedy následující rovnice:

$$\ddot{x}_1 = g - \frac{k_0}{m_1(\phi r - \pi r)}(x_1 - \phi r + \pi r) - \frac{b_0}{m_1(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) - \frac{F_{f1}}{m_1} \quad (61)$$

$$\ddot{x}_2 = g - \frac{k_0}{m_2(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0}{m_2(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - \frac{F_{f2}}{m_2} + a_e \quad (62)$$

$$\begin{aligned} \ddot{\phi} = & \frac{T}{J} + \frac{k_0 r}{J(\phi r - \pi r)}(x_1 - \phi r + \pi r) + \frac{b_0 r}{J(\phi r - \pi r)}(\dot{x}_1 - \dot{\phi} r) + \dots \\ & \dots - \frac{k_0 r}{J(l_0 - \phi r)}(x_2 - l_0 + \phi r) - \frac{b_0 r}{J(l_0 - \phi r)}(\dot{x}_2 + \dot{\phi} r) - b_f \frac{\dot{\phi}}{J} \end{aligned} \quad (63)$$

ze kterých lze linearizací získat následující MIMO systém:

$$\mathbf{z} = [z_1, z_2, z_3, z_4, z_5, z_6] = [x_1, x_2, \phi, \dot{x}_1, \dot{x}_2, \dot{\phi}] ; \mathbf{u} = [T, a_e]^T$$

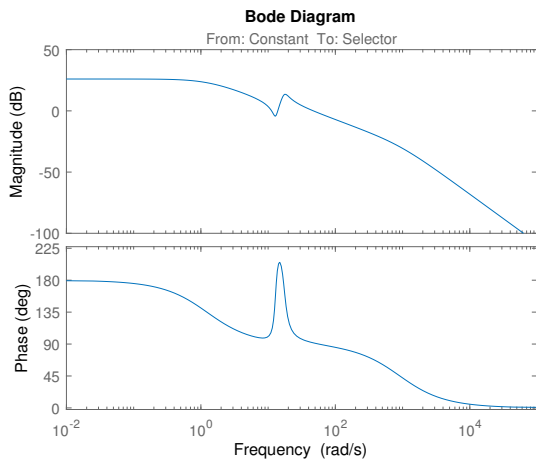
$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{-k_0}{m_1 l_1} & 0 & \frac{k_0 r z_1}{m_1 l_1^2} & \frac{-b_0}{m_1 l_1} & 0 & \frac{b_0 r}{m_1 l_1} \\ 0 & \frac{-k_0}{m_2 l_2} & \frac{-k_0 r z_2}{m_2 l_2^2} & 0 & \frac{-b_0}{m_2 l_2} & \frac{-b_0 r}{m_2 l_2} \\ \frac{k_0 r}{J l_1} & \frac{-k_0 r}{J l_2} & \frac{-k_0 r^2 (z_2 l_1^2 + z_1 l_2^2)}{J l_1^2 l_2^2} & \frac{b_0 r}{J l_1} & \frac{-b_0 r}{J l_2} & \frac{-r^2 (l_1 + l_2 - b_f l_1 l_2) b_0}{J l_1 l_2} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ \frac{1}{J} & 0 \end{bmatrix} ; \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} ; \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

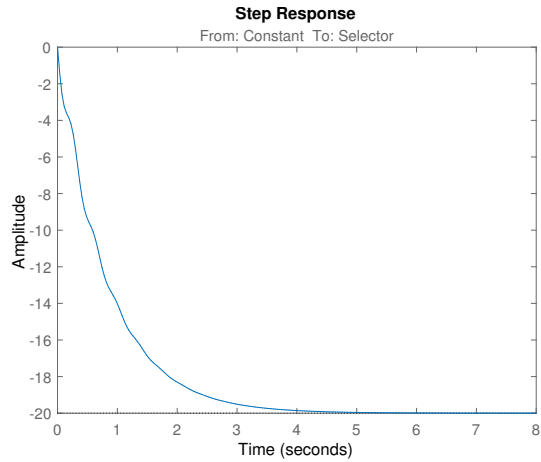
$$\dot{\mathbf{z}}(t) = \mathbf{A}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{z}(t) + \mathbf{D}\mathbf{u}(t) \quad (64)$$

Tento systém zlinearizujeme pro 3 polohy: 3m, 15m a 30m. Tím získáme představu o působení poruchy v jednotlivých výškách. Následně v Simulinku poskládáme MIMO schéma (obrázek 79) a využijeme aplikaci Model Linearizer k získání přenosu z působící poruchy na rychlost motoru za současného působení robustního regulátoru. Pro tyto přenosy pak vykreslíme Bodeho, přechodové a impulsní charakteristiky. Impulsní charakteristiky vykreslíme rovněž pro přenos z poruchy na rychlost kabiny (taktéž za současného působení robustního regulátoru).

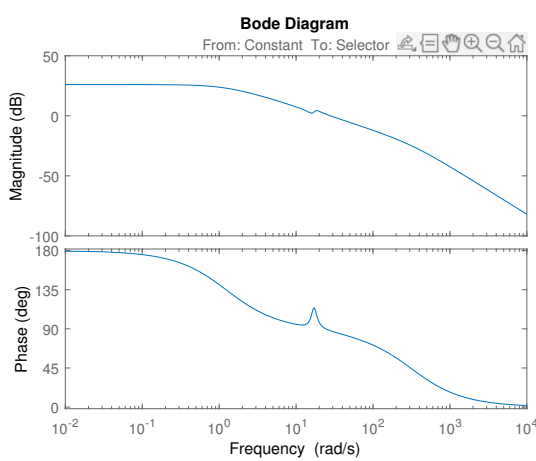


(a) Bodeho charakteristika

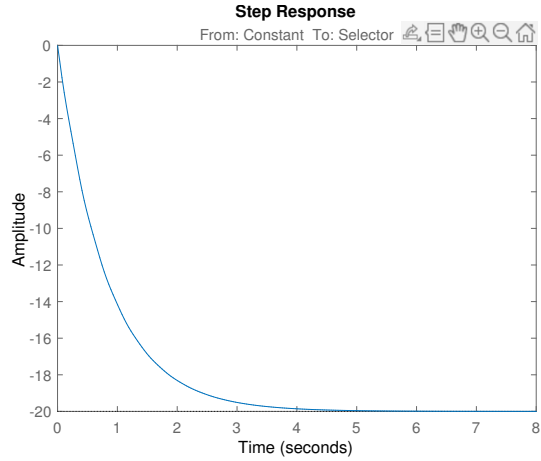


(b) Přebodová charakteristika

Obrázek 71: Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 3m$

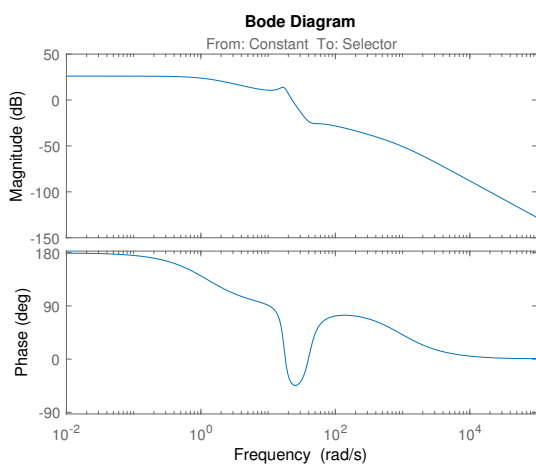


(a) Bodeho charakteristika

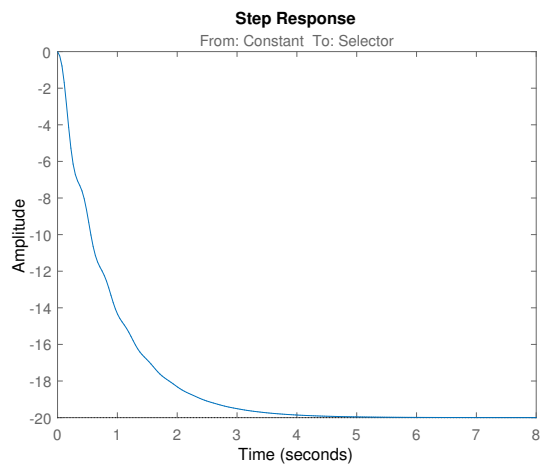


(b) Přebodová charakteristika

Obrázek 72: Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 15m$

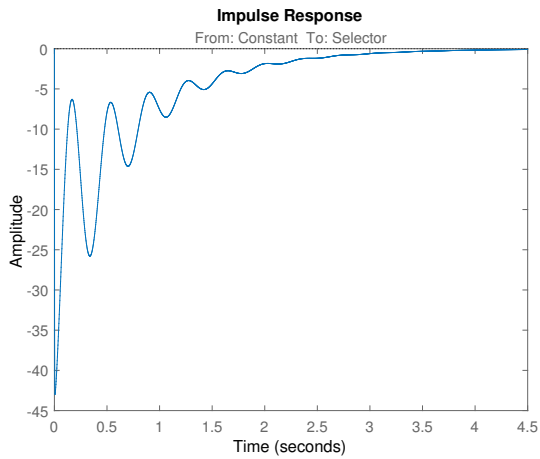


(a) Bodeho charakteristika

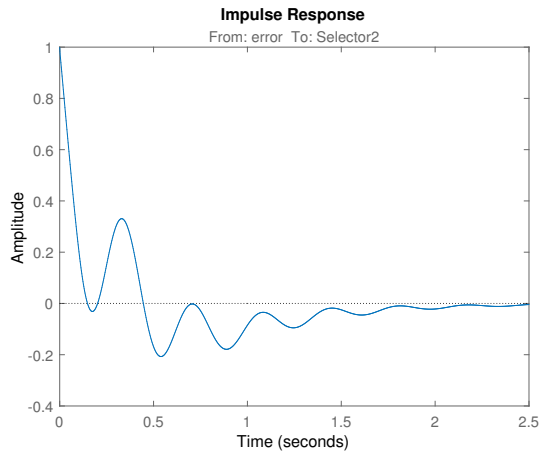


(b) Přebodová charakteristika

Obrázek 73: Přenos z poruchy na rychlost motoru - linearizovaný v $x_2 = 30m$

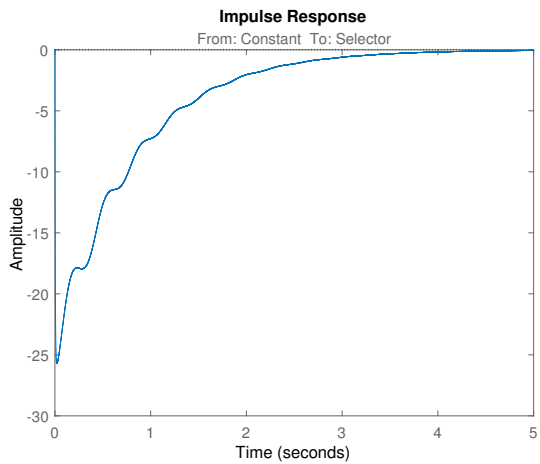


(a) Přenos na motor

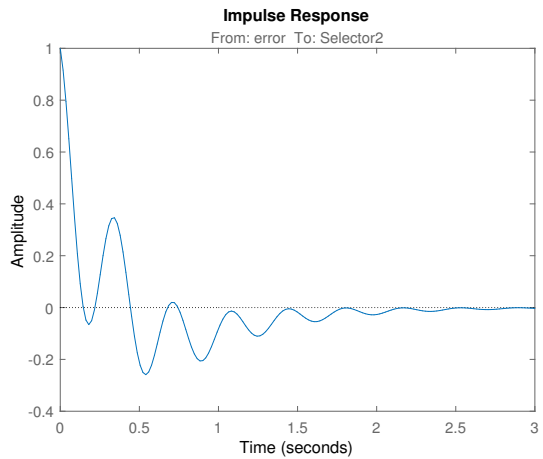


(b) Přenos na kabinu

Obrázek 74: Impulsní charakteristiky - přenosy linearizované v $x_2 = 3m$

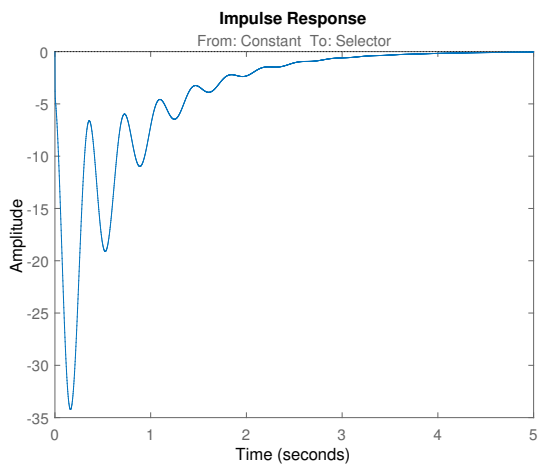


(a) Přenos na motor

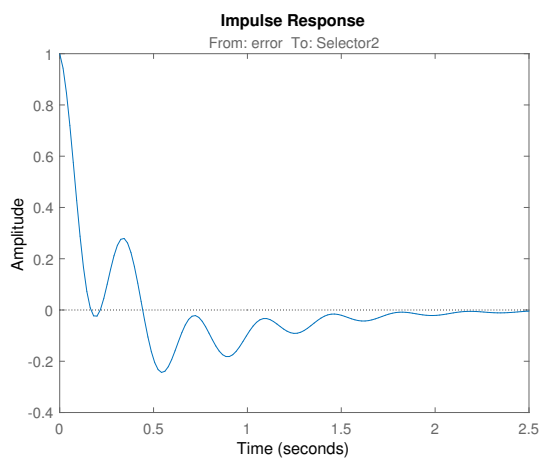


(b) Přenos na kabinu

Obrázek 75: Impulsní charakteristiky - přenosy linearizované v $x_2 = 15m$

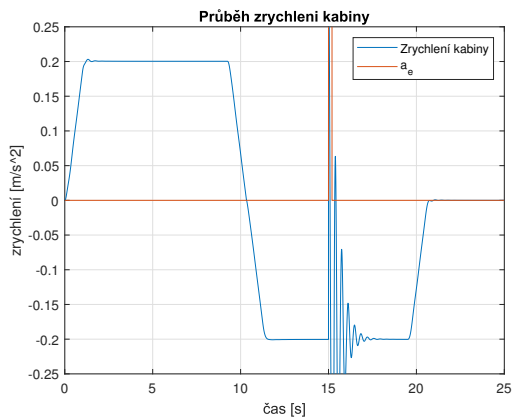


(a) Přenos na motor

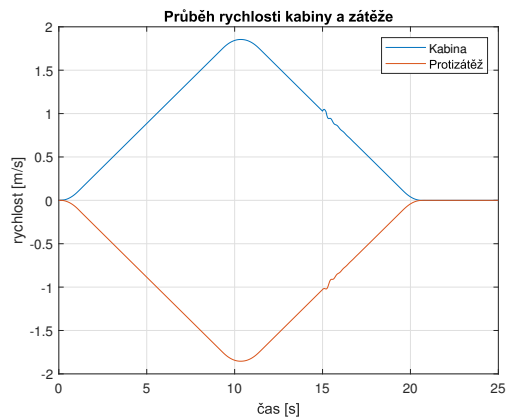


(b) Přenos na kabinu

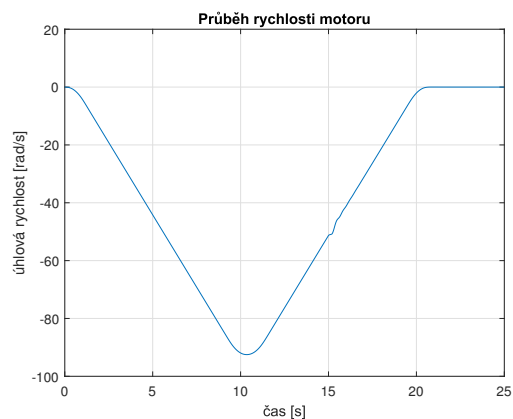
Obrázek 76: Impulsní charakteristiky - přenosy linearizované v $x_2 = 30m$



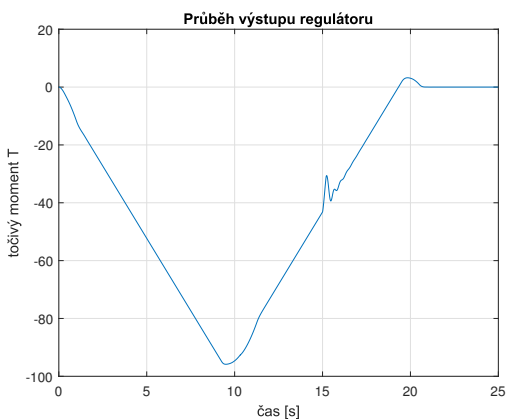
(a) Zrychlení kabiny



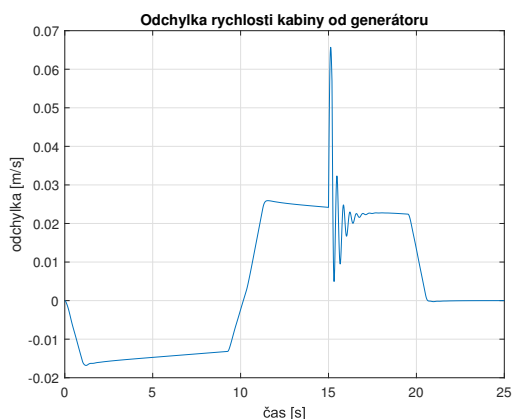
(b) Rychlost kabiny a protizátěže



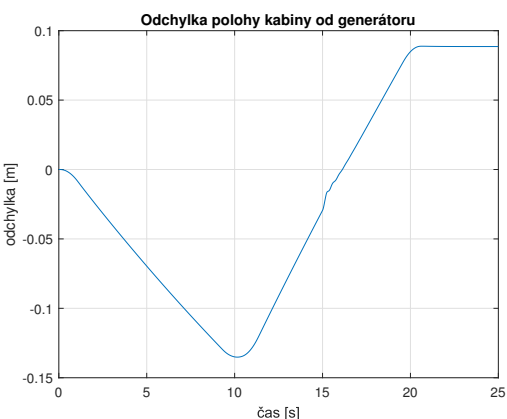
(c) Rychlost motoru



(d) Výstup regulátoru

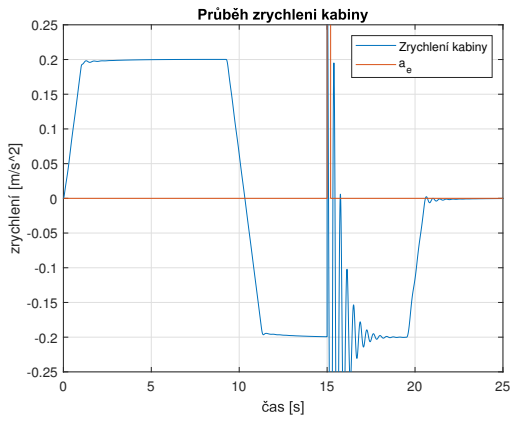


(e) Odchylka rychlosti kabiny od pož.

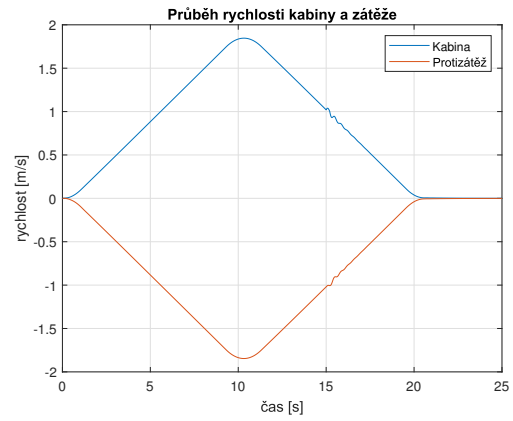


(f) Odchylka polohy kabiny od pož.

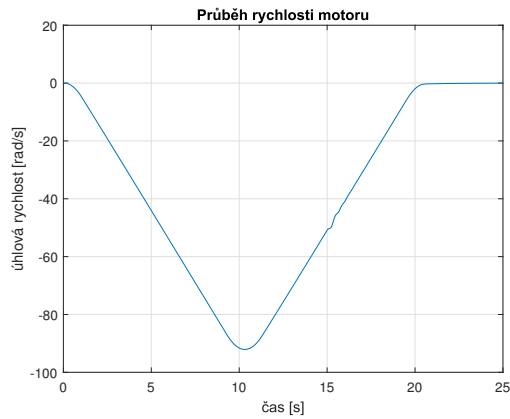
Obrázek 77: Simulace regulace dvěma regulátory s poruchou $0.6ms^{-2}$ působící v čase $t = 15s$



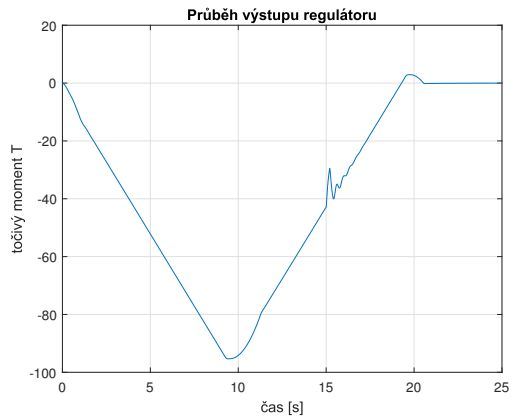
(a) Zrychlení kabiny



(b) Rychlost kabiny a protizátěže

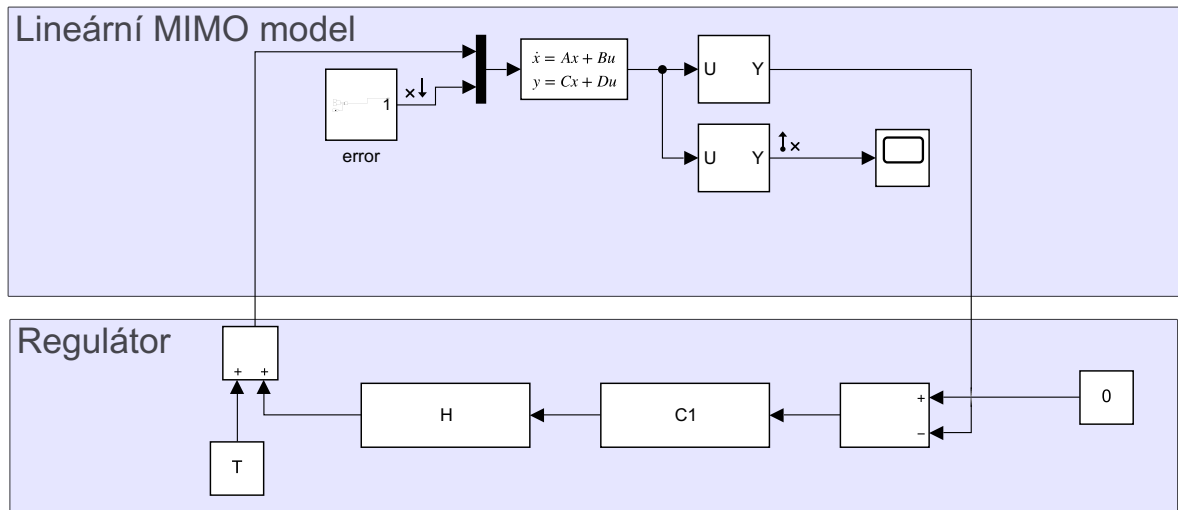


(c) Rychlost motoru



(d) Výstup regulátoru

Obrázek 78: Simulace regulace DK s poruchou $0.6ms^{-2}$ působící v čase $t = 15s$



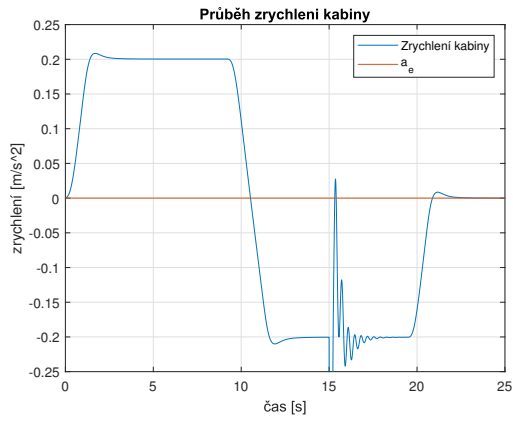
Obrázek 79: Schéma mimo systému použitého pro získání přenosů z působící chyby a_e

Potvrzuje se tedy, že aby bylo možno takovéto chyby efektivně regulovat, byly by potřeba ještě další sledovatelné stavy, například rychlost kabiny nebo ideálně samotné zrychlení kabiny (zde by postačovala i dostatečně přesná rekonstrukce), protože impulsní charakteristiky obou přenosů (obrázky 74, 75, 76) se ustalují zhruba 3 sekundy, a to za současného působení robustní regulace. Z dalších simulací je pak zjevné, že ostatní druhy regulace (regulace dvěma regulátory - obrázek 77 - a také dynamický kompenzátor - obrázek 78) též neposkytují nikterak zásadně lepší regulaci poruchy (u těch nelze udělat charakteristiky, jelikož využívají ke svému fungování Matlab funkci, což nedokáže linearizační aplikace reflektovat), tudíž ustálení zrychlení kabiny lze přisuzovat spíše samotnému tlumení lana b , než regulaci.

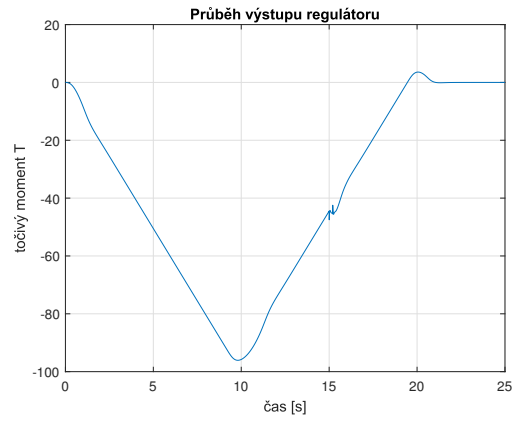
U regulace dvěma regulátory navíc ještě dochází k trvalému posunu odchylky polohy (o $\pm 8\text{cm}$, což již považujeme za problémové) od generátoru trajektorie - stejně jako bez působení poruchy, viz obrázek 51 v sekci 3.4. V tomto ohledu je významně lepší volnou DK, jelikož ten svou SZV tlačí odchylku vždy do nuly. Jak jsme již řekli, jedním z možných řešení pro GS by bylo zavedení vnější polohové smyčky, která by tento nedostatek kompenzovala.

8.2 Vstupní porucha

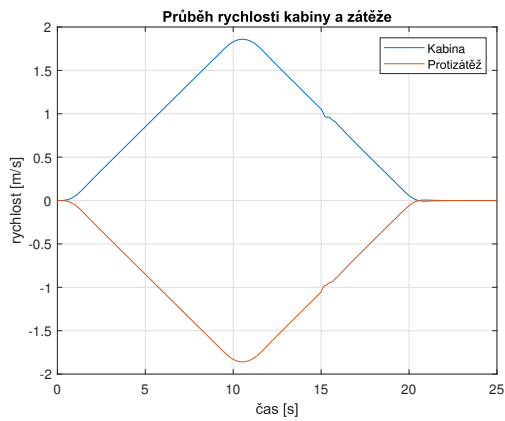
Protože výsledky předchozí sekce, věnované vnitřní poruše a_e , nemusí působit úplně uspokojivě, provedeme simulace ještě za působení vstupní poruchy (tz. poruchy působící za regulátorem přímo na točivý moment motoru T). Tuto poruchu necháme působit vždy v čase $t = 15\text{s}$, přičemž trvá 0.2s a má hodnotu 10Nm^{-1} . Z výsledků simulací - obrázky 80, 81 a 82 - je pak vidět, že u této chyby se zásahy regulací liší. Jako nejlepší se pak jeví dynamický kompenzátor, protože značně omezuje oscilace rychlosti kabiny vyvolané vstupní chybou a tedy i kmity v jejím zrychlení. Také dosahuje významně nižších odchylek rychlosti i polohy oproti požadované trajektorii.



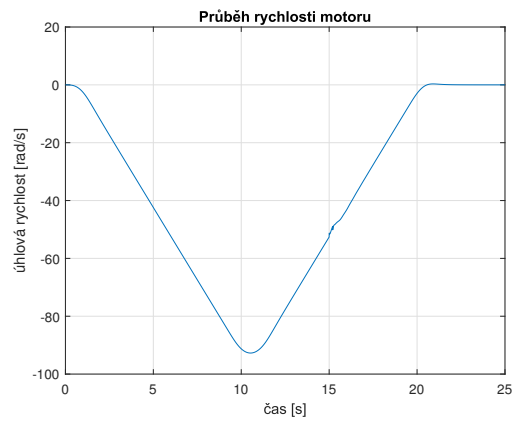
(a) Zrychlení kabiny



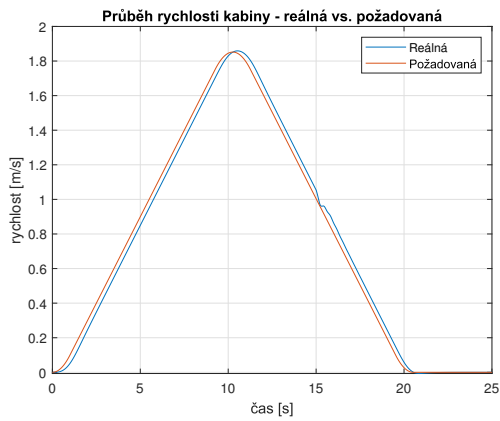
(b) Výstup regulátoru



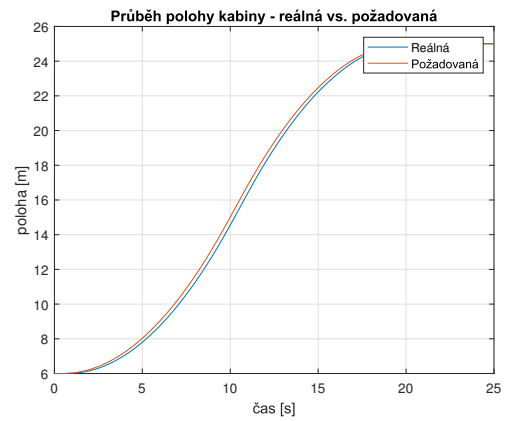
(c) Rychlost kabiny a protizátěže



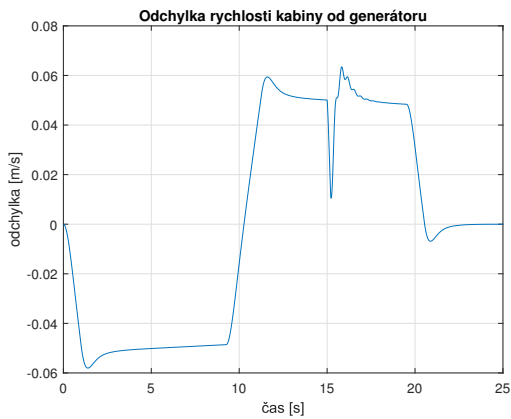
(d) Rychlost motoru



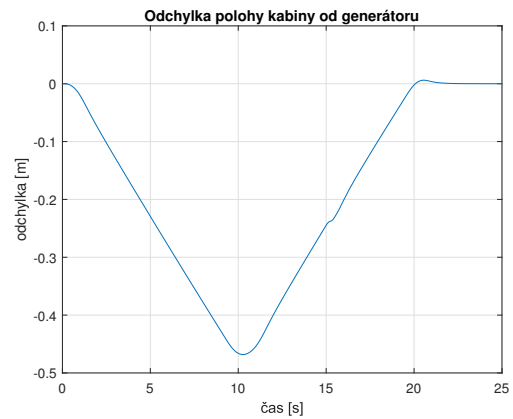
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná

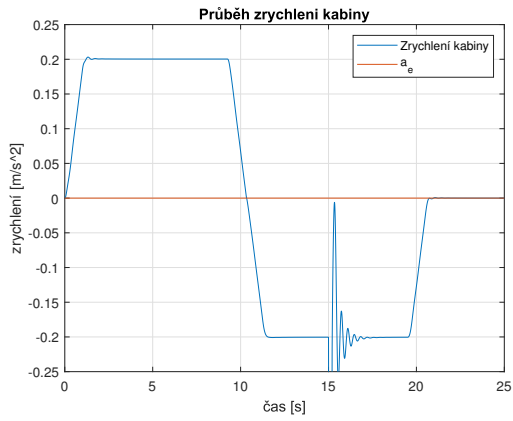


(g) Odchyłka rychlosti kabiny od pož.

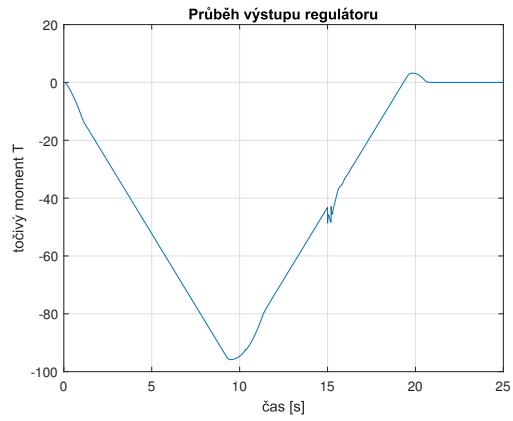


(h) Odchyłka polohy kabiny od pož.

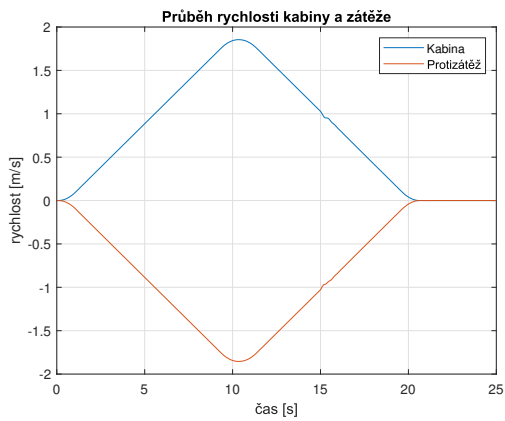
Obrázek 80: Simulace robustní regulace se vstupní poruchou



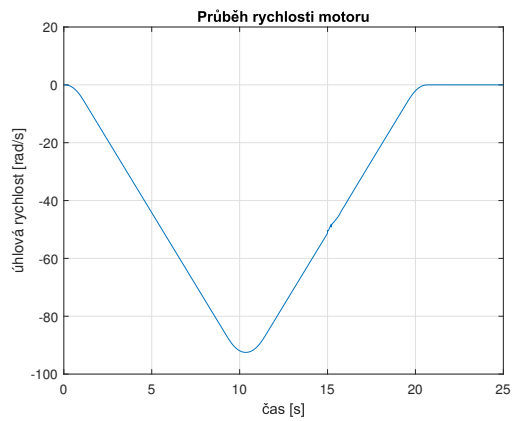
(a) Zrychlení kabiny



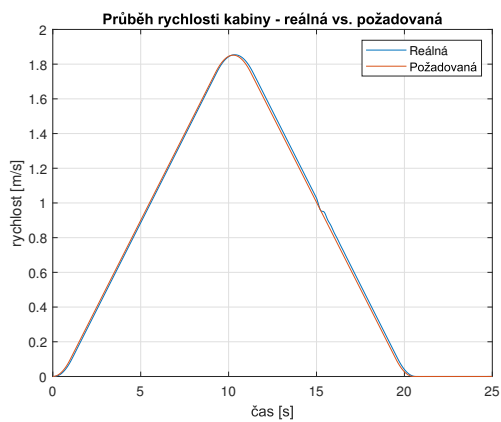
(b) Výstup regulátoru



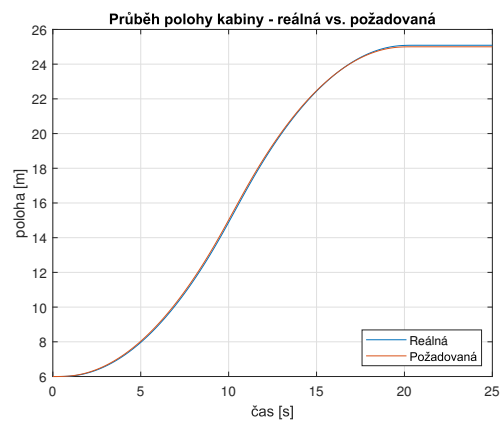
(c) Rychlost kabiny a protizátěže



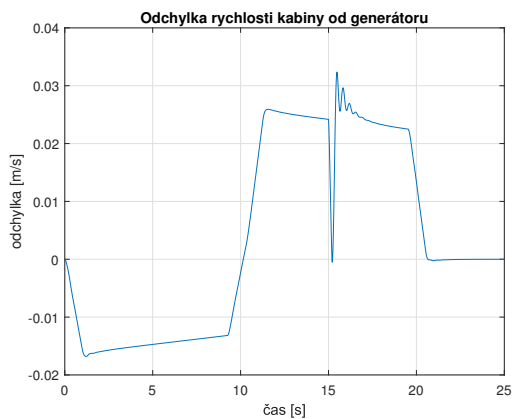
(d) Rychlost motoru



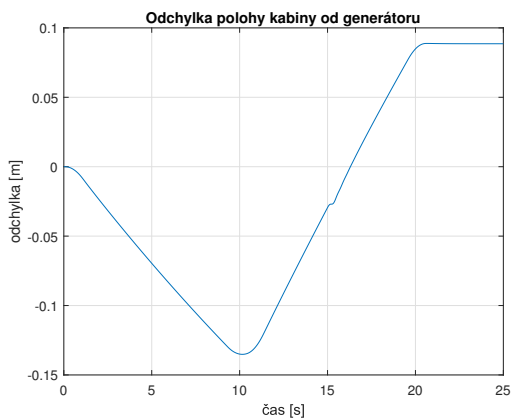
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná

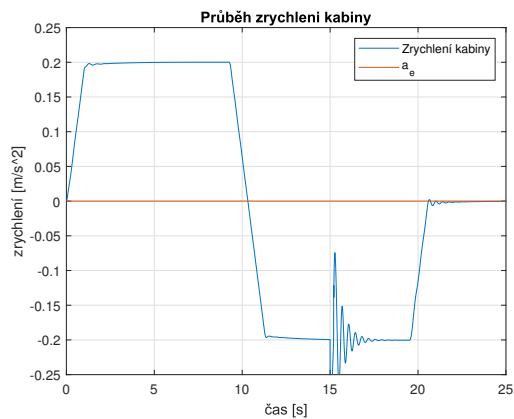


(g) Odchylka rychlosti kabiny od pož.

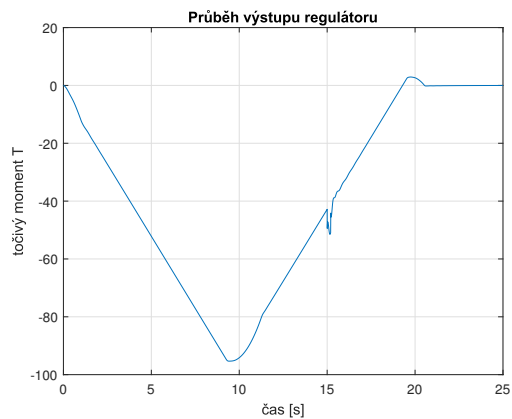


(h) Odchylka polohy kabiny od pož.

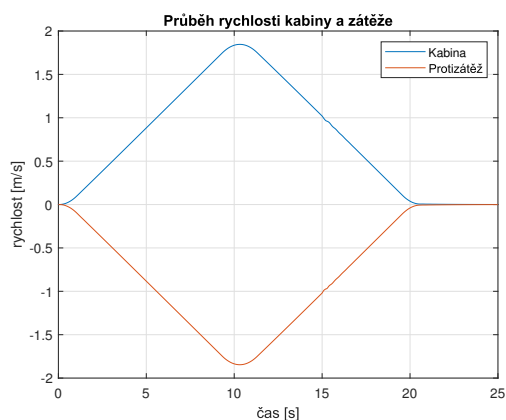
Obrázek 81: Simulace regulace dvěma regulátory se vstupní poruchou



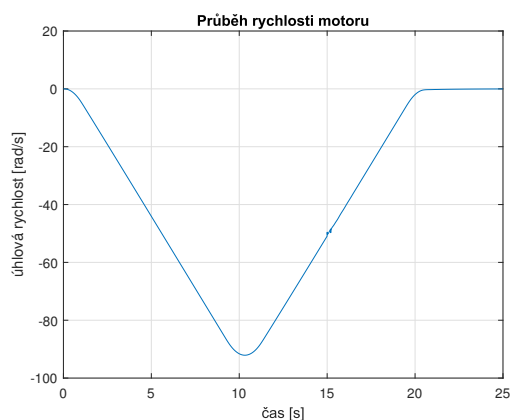
(a) Zrychlení kabiny



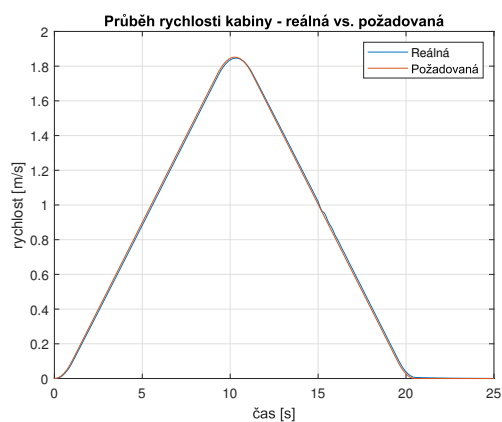
(b) Výstup regulátoru



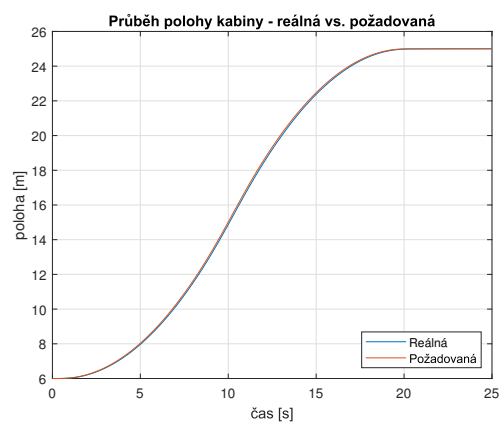
(c) Rychlost kabiny a protizátěže



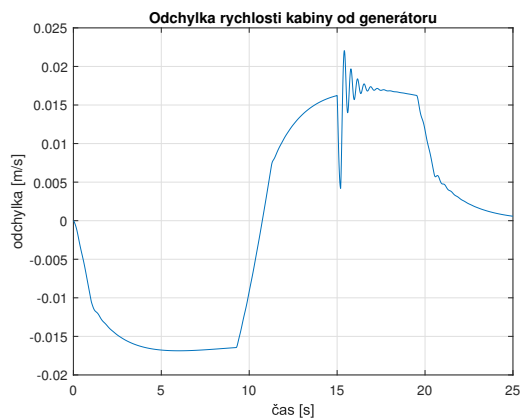
(d) Rychlost motoru



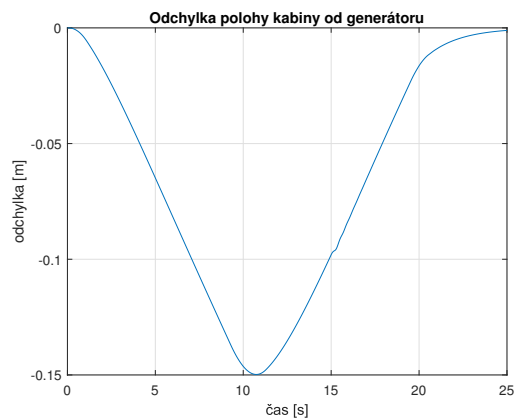
(e) Rychlost kabiny pož. vs. reálná



(f) Poloha kabiny pož. vs. reálná

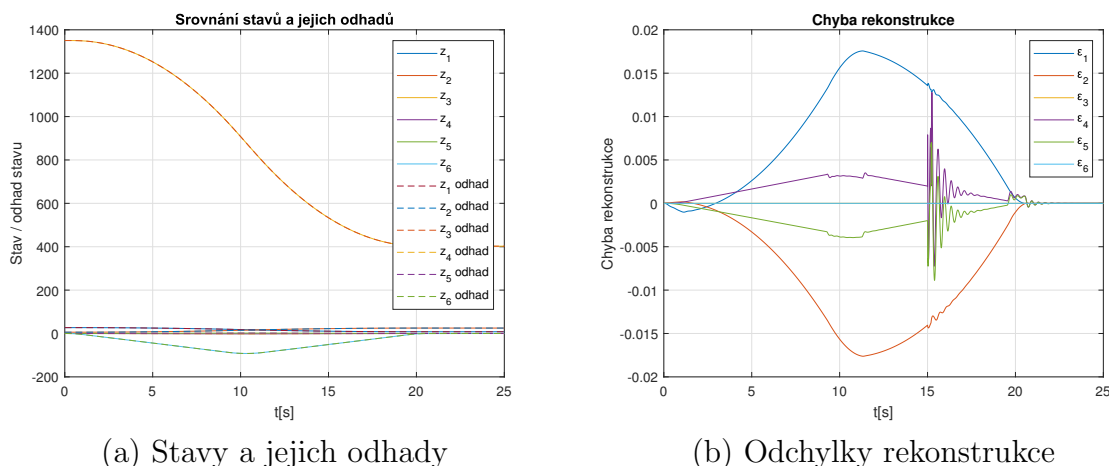


(g) Odchylna rychlosti kabiny od pož.



(h) Odchylna polohy kabiny od pož.

Obrázek 82: Simulace dynamického kompenzátoru se vstupní poruchou



Obrázek 83: Rekonstrukce DK při simulaci se vstupní poruchou

9 Závěrečné srovnání zkoumaných strategií řízení

Na závěr uvedeme přímé srovnání jednotlivých metod řízení. Pro přehlednost budeme srovnávat pouze nejlepší variantu GS - dva regulátory, pak hybridní SDRE-PI DK a původní robustní regulátor.

Uvedeme jak porovnání jednotlivých trajektorií včetně generátoru (obrázek 84), tak odchylky stavů od požadované trajektorie (obrázek 85).

Dále pak uvedeme tabulku 2 obsahující maximální odchylky stavů od požadované trajektorie, ISE kritéria jednotlivých stavů (odchylka je brána od požadované trajektorie) a nakonec dobu trvání manévru (zde uvažujeme pohyb z počáteční polohy x_{20} do blízkosti koncové $x_{21} \pm \varepsilon$, kde zvolíme $\varepsilon = 3mm$ - v této blízkosti se již aktivuje manuální brzda a výtah zastaví).

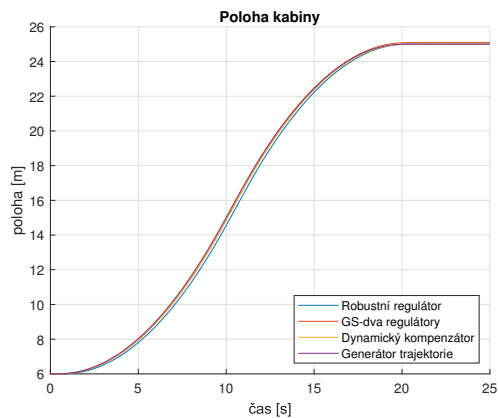
U doby dojezdu při PI regulaci musíme uvažovat, že je takto nízká, protože poloha má překmit. To ale v realitě není možné, protože výtah by narazil na mechanické dojezdové spínače a zastavil. Výtah by při této regulaci tedy spustil manuální brzdu v moment, kdy má ještě poměrně vysokou rychlost, což by jistě nebylo komfortní.

*ISE kritérium polohy je v případě GS značně zavádějící, jelikož správně má tvar:

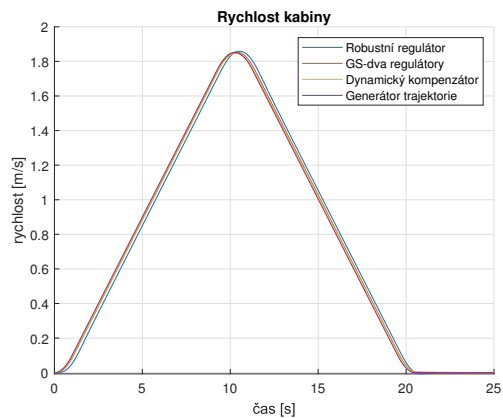
$$ISE = \int_0^{\infty} e^2(t) dt \quad (65)$$

tedy je počítáno od nuly do nekonečna. My jej však počítáme pouze do doby konce simulace. Jelikož je koncová odchylka nenulová (systém má jinou koncovou polohu než je požadovaná), musí být ISE kritérium v tomto případě nekonečné, $ISE_{GS} = \infty$.

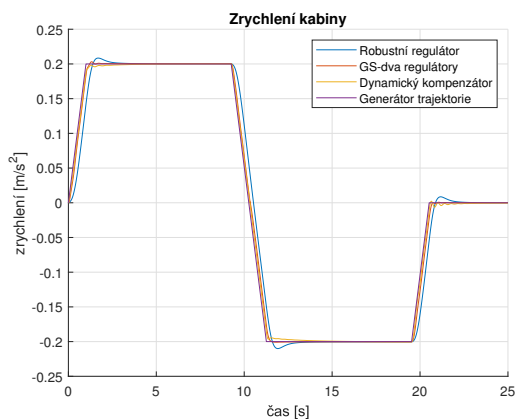
Ze stejného důvodu je pak nekonečná i doba manévru.



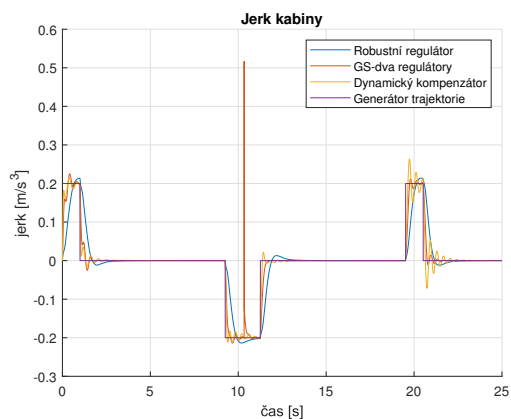
(a) Poloha



(b) Rychlost

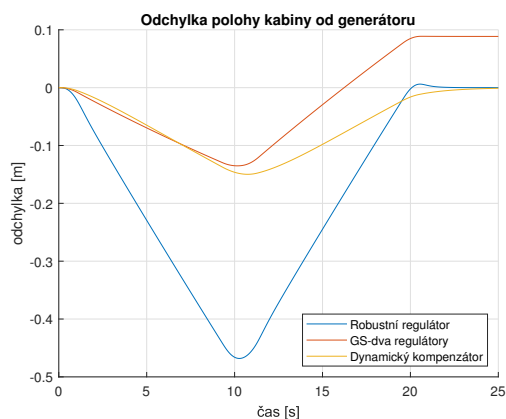


(a) Zrychlení

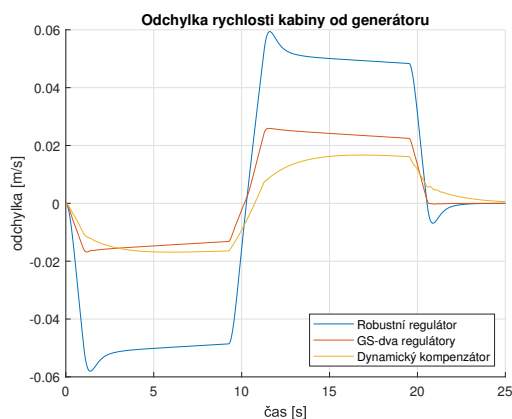


(b) Jerk

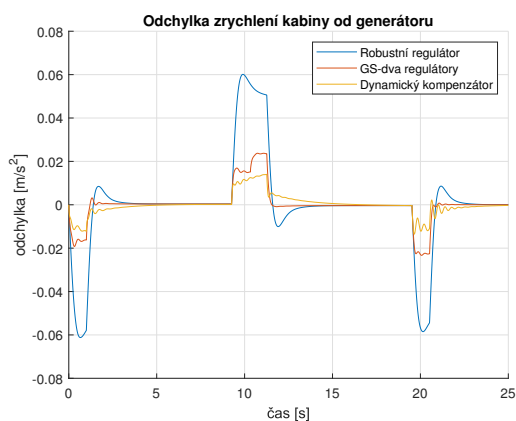
Obrázek 84: Srovnání trajektorie s GT



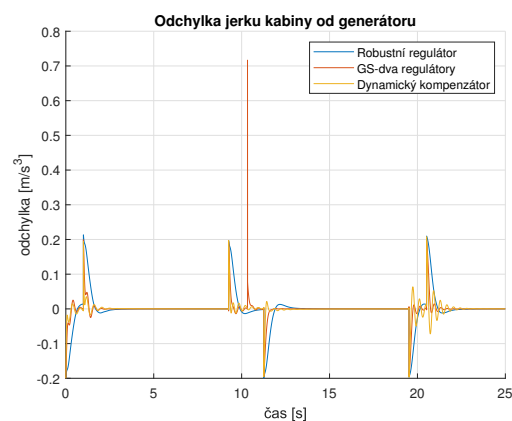
(a) Odchylka polohy



(b) Odchylka rychlosti



(a) Odchylka zrychlení



(b) Odchylka jerku

Obrázek 85: Odchytky jednotlivých regulací oproti GT

Kritéria kvality regulace:

	Robustní regulátor	GS - dva regulátory	Dynamický kompenzátor
Max. odchylka polohy	0.4681	0.1352	0.1498
Max. odchylka rychlosti	0.0594	0.0259	0.0167
Max. odchylka zrychlení	0.06121	0.02375	0.01404
Max. odchylka jerku	0.21325	0.71673	0.20781
ISE polohy	1538.7721	159.2378*	172.4619
ISE rychlosti	46.3774	7.1235	4.3738
ISE zrychlení	11.43314	1.50428	0.56264
ISE jerku	52.22761	15.84582	9.93505
Doba manévru	19.976	∞ *	23.069

Tabulka 2: Kritéria kvality regulace

10 Repetitive control

Tuto metodu řízení jsme nakonec neimplementovali, jelikož není k tomuto účelu vhodná. Je totiž určena k vylepšení řízení v případech, kdy je referenční hodnota nebo působící porucha periodický signál.

Jde o metodu založenou na principu vnitřního modelu, která užívá informace z předchozích period (cyklů) k modifikaci řídicí veličiny za účelem dosažení asymptotického sledování periodického referenčního signálu nebo potlačení periodické poruchy.^[10]

Naše reference ale není periodická a žádnou periodicky působící chybu neuvažujeme. Není tedy důvod tuto metodu využít.

Z této metody pak vychází metoda *iterative learning control*. V jejím případě probíhá simulace (například pohybu po dané trajektorii) opakovaně a na základě předchozích se vylepšují parametry řídicího systému té současně.

11 Test na reálném systému

Z důvodu velkého rozsahu této práce jsme se rozhodli od testu na reálném systému upustit. Bylo by potřeba systém znovu identifikovat (jelikož jeho dynamika nebude shodná s námi uvažovaným teoretickým modelem), následně znovu navrhnout všechny metody řízení, otestovat a vykreslit všechny grafy (bez i s poruchou). Práce by pak byla neúměrně rozsáhlá, čemuž se chceme vyvarovat.

^[10]KKY/NŘS

Závěr

V naší práci jsme se zaměřili na vylepšení parametrů regulace výtahů oproti základnímu robustnímu PI regulátoru. Při tom jsme využívali čistě zpětnovazební řízení, jelikož metodám přímovazebního řízení se věnuje ve své práci kolega Jakub Tvrz[10].

Metoda gain-scheduling se ukázala jako vhodná vzhledem k tomu, že umíme sledovat polohu kabiny. Jako nejvýhodnější ze všech variant gain-schedulingu se ukázala metoda dvou regulátorů přepnutých uprostřed jízdy výtahu, protože je jednoduchá a dosahuje velmi dobrých výsledků. Jejím negativem je ale velký ráz jerku při přepnutí regulátorů a také nenulová koncová odchylka (to lze však vyřešit zavedením vnější polohové smyčky).

Následně jsme díky linearizaci podél trajektorie mohli vytvořit časově variantní stavovou zpětnou vazbu ve formě SDRE, která ve spojení se základním robustním PI regulátorem dosahuje ještě lepších výsledků (zejména rychleji sleduje referenci), ovšem za cenu významně složitější struktury regulace.

Jako další možné vylepšení regulace vidíme 'gain-scheduling rekonstruktorů', protože druhý rekonstruktor v místě dojezdu by mohl pomoci značně utlumit dojezdové kmity. Další variantou by bylo využití pokročilé metody *iterative learning control*, která měla být původně součástí této bakalářské práce, ovšem ukázala se jako příliš složitá (hlavně z pohledu teorie), a tudíž jsme na doporučení vedoucího práce raději linearizovali systém podél trajektorie a implementovali LQR.

Pro potlačení oscilací zrychlení kabiny vyvolaných externí poruchou (například pohybem lidí v kabině) by pak byl potřeba buď přímo senzor zrychlení kabiny, nebo přídatné tlumiče mezi kabinou a lanem.

Reference

- [1] Břetislav Kubeš and Jakub Tvrz. KKY/PAŘR5: Robustní řízení nelineárního modelu výtahu ve vícero rovnovážných bodech, KKY FAV ZČU, 2023.
- [2] Toshiba: the technology of providing riding comfort to passengers - car system. <https://www.toshiba-elevator.co.jp/elv/infoeng/technology/fastest/tec02.html>. Accessed: 2023-05-15.
- [3] Miloš Schlegel, Martin Goubej, and Jana Königsmarková. Active vibration control of two-mass flexible system using parametric jordan form assignment. *IFAC Proceedings Volumes*, 45:477–482, 12 2012.
- [4] Martin Goubej. Fundamental performance limitations in pid controlled elastic two-mass systems. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 828–833, 07 2016.
- [5] Michal Špirk. Automatické ladění regulátorů pro elektrické servopohony. Bachelor's thesis, KKY FAV ZČU, 2021.
- [6] Michal Špirk. H_∞ pro elektromechanické soustavy: Uživatelská příručka, KKY FAV ZČU, 2022.
- [7] Marian Blejan and Robert Blejan. Mathematics for real-time s-curve profile generator. “*HIDRAULICA*” (No. 4/2020) *Magazine of Hydraulics, Pneumatics, Tribology, Ecology, Sensorics, Mechatronics - ISSN 1453 – 7303*, pages 7–25, 12 2020.
- [8] Wolfgang Sienel and Tilman Bunte. Design of gain scheduling controllers in parameter space. In *1997 European Control Conference (ECC)*, pages 553–556, 1997.
- [9] Tayfun Çimen. State dependent riccati equation (sdre) control: A survey. *Plenary Session of 17th IFAC World Congress*, 17, 07 2008.
- [10] Jakub Tvrz. Přímovazební řízení pro kompenzaci nežádoucích vibrací kabiny výtahů. Bachelor's thesis, KKY FAV ZČU, 2023.