ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

Stabilizace pohybu mostového jeřábu při manuálním řízení

Gantry crane motion stabilization under manual control

Bakalářská práce

Michal Verner

Vedoucí práce: Ing. Petr Havel, Ph.D. Praha, 2010

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze, dne …….

…….
podpis

Poděkování

Děkuji Ing. Petru Havlovi, Ph.D. za pomoc, strávený čas a cenné rady při realizaci této bakalářské práce. Děkuji také firmě SIMO-CZ, s.r.o. za možnost řešit takto zajímavé téma.

Abstrakt

Tato práce se zabývá přímovazebním řízením mostového jeřábu při manuálním ovládání operátorem. Cílem řízení je minimalizovat výchylku břemene a zároveň zajistit operátorovi příjemnou obsluhu, aby nedocházelo k tomu, že se bude operátor s řídicím systémem „přetahovat“. Metody, které jsou v této práci použity, mají za úkol natvarovat vstupní signál tak, aby došlo k co možná nejmenšímu výkyvu břemene při přejezdu jeřábu. První metodou je input shaping, druhou je filtrování. U input shapingu jsou uvedeny tři druhy shaperů, každý z nich má různé vlastnosti, co se týče rychlosti náběhu systému a robustnosti při změně parametrů. Každý z těchto shaperů je otestován na matematickém modelu jeřábu. U filtrování je využito dolních propustí a pásmových zádrží, aby se do systému nedostávala energie na frekvencích budící kmitavé módy soustavy. Nakonec jsou obě metody porovnány.

Vlastní řídicí systém je navržen na modelu, který je možno obsluhovat pomocí joysticku ve virtuální realitě. Návrh je proveden jen pomocí jednoduchých operací, aby se mohl snadno implementovat do PLC.

V závěru práce jsou představeny výsledky ukazující, že řídicí systém zvládne efektivně potlačit zbytkové kývání břemene a umožňuje operátorovi snadněji a bezpečně ovládat mostový jeřáb. Navrhovaný systém by mohl být implementován do reálného řízení mostového jeřábu.

Abstract

This thesis presents the feed forward control of gantry crane under manual control. The goal of the control system is to minimize the swing of the load and to provide an operator friendly application. There are two control methods mentioned in this work. Both of them shape the input to reduce the output vibrations during movement of the crane. First method is called input shaping. There are three types of input shapers in this thesis. Every type has different properties and they differ in the robustness too. All shapers are tested on a mathematical model of gantry crane. The second method is a filtering technique. The filters do not pass energy at specific frequencies to the system. This eliminates the swing of the load, because oscillations modes are suppressed. Finally both methods are compared.

The new control system is designed for the model in virtual reality. It is possible to control the model of the crane via joystick. The control system is designed by using simple operations. This helps to easy implementation to PLC.

At the end of the work are presented results showing that the control system reduces residual swing of the load and helps operator to control the crane safely and easily. Designed system can be implemented into the real gantry crane.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Michal Verner**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Stabilizace pohybu mostového jeřábu při manuálním řízení**

Pokyny pro vypracování:

1. Seznamte se s problematikou řízení pohybu mostových jeřábů
2. Navrhněte vhodný regulační systém, který pomůže operátorovi jeřábu při ručním řízení udržet rozkmit břemene v žádoucích mezích
3. Otestujte funkci systému na simulačním modelu reálného jeřábu

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Petr Havel, Ph.D.

Platnost zadání: do konce zimního semestru 2010/2011

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 30. 10. 2009

Obsah

1 ÚVOD 11

1.1 ZADÁNÍ A POPIS PROBLÉMU 11

1.1.1 Popis Mostového jeřábu 11

1.2 NAVRHOVANÉ ŘEŠENÍ 12

2 MOŽNOSTI PŘÍMOVAZEBNÍHO ŘÍZENÍ 13

2.1 REŠERŠE 13

2.2 INPUT SHAPING 13

2.2.1 Odvození input shapingu pro lineární systém 15

2.2.2 Druhy input-shapingu 16

2.3 FILTROVÁNÍ 19

3 MODEL MOSTOVÉHO JEŘÁBU 20

3.1 DIFERENCIÁLNÍ ROVNICE SOUSTAVY 20

3.1.1 Linearizace systému 22

3.2 IDENTIFIKACE MODELU 23

4 APLIKACE ŘÍZENÍ NA MODELU 27

4.1 INPUT SHAPING 27

4.1.1 ZV-shaper 28

4.1.2 ZVD-shaper 30

4.1.3 EI-shaper 32

4.2 FILTROVÁNÍ 34

4.2.1 FIR – dolní propust 34

4.2.2 FIR – pásmová zádrž 35

4.2.3 IIR filtry 36

4.3 POROVNÁNÍ OBOU METOD ŘÍZENÍ 36

5 APLIKACE ŘÍZENÍ NA REÁLNÉM JEŘÁBU 38

5.1 BEZPEČNOSTNÍ OPATŘENÍ 38

5.2 IMPLEMENTACE ŘÍDICÍHO SYSTÉMU 40

5.2.1 Blok zpracování setpointu rychlosti a bezpečnostních opatření 40

5.2.2 Blok generování rychlosti 43

5.2.3 Blok tvořící shaper a rozhodující o aplikaci input shapingu 44

5.3 TESTY ŘÍDICÍHO SYSTÉMU 46

5.3.1 Robustnost systému vůči neznámé délce závěsu 46

5.3.2 Test užitečnosti a funkčnosti celého systému 47

6 ZÁVĚR 49

7 CITOVANÁ LITERATURA 50

8 PŘÍLOHY 51

1 Úvod

Tuto práci jsem si vybral, protože mě zajímá řídicí technika a teorie řízení. Zároveň jsem chtěl zpracovat téma, které má reálnou aplikaci.

1.1 Zadání a popis problému

Cílem práce bylo seznámit se s problematikou řízení pohybu mostových jeřábů, navrhnout vhodný regulační systém, který pomůže operátorovi udržet při ručním řízení rozkmit břemene v žádoucích mezích a otestovat funkci systému na simulačním modelu reálného jeřábu.

Zadavatelem práce je firma SIMO-CZ. Mostové jeřáby řízené systémy této firmy jsou používány ve slévárnách, překladištích, skladech, v jaderné elektrárně v Dukovanech a v mnoha dalších objektech. Příkladem takového jeřábu je jeřáb ve skladu kamení v Častolovicích:

Obr. 1.1: Mostový jeřáb v Častolovicích

1.1.1 Popis Mostového jeřábu

Jeřáb, na kterém má být řízení vyzkoušeno, je právě jeřáb ve skladu kamení v Častolovicích, viz Obr. 1.1. Celková hmotnost jeřábu je *20,9 t*. Nosnost jeřábu *je 6,3 t*. Samotný jeřáb se skládá ze tří částí – drapák, jeřábová kočka a most. Drapák je ta část, do které se nabírá náklad. Drapák je zdvihán na laně pomocí motoru, který má výkon *30 kW*. Rychlost zdvihání

nebo spouštění je *2,5 – 25 m/min*. Kočka je část, na které je připevněna kabina s operátorem a visící drapák. Kočka zajišťuje pohyb břemene v jednom směru. Dva motory, které pohybují kočkou, mají každý výkon *1,10 kW* a kočka se díky nim pohybuje rychlostí v rozmezí *5 – 50 m/min*. Most je část, po které se pohybuje kočka a sám most se pohybuje po kolejnicích ve směru kolmém na pohyb kočky. Rozpětí mostu je *20,10 m*. Most je poháněn dvěma motory, každý má výkon *5,5 kW*. Díky nim se most pohybuje rychlostí *8 – 80 m/min*. Délka kolejnic, po kterých se pohybuje most, je *133 m*.

Samotné motory jsou řízené frekvenčními měniči, které umožňují nastavit relativně přesně požadovanou rychlost jednotlivých motorů. Samotné měniče měří řadu veličin, jako je například moment, proud protékající vinutím atd. Další měřenou veličinou je hmotnost nákladu. Délka lana není měřena, ale není problém tento senzor do soustavy přidat.

Z důvodů bezpečnosti je jeřábu omezena rychlost, když se dostane moc blízko ke stěnám haly. Takto je celá oblast, kde se pohybuje jeřáb, rozdělena do tří zón. Více informací o zónách viz kapitola 5.1.

V současné době je jeřáb řízen operátorem pomocí čtyřpolohového joysticku, tímto joystickem operátor zadává do řídicího systému čtyři různé setpointy rychlosti: 10%, 25%, 50% a 100% maximální rychlosti, jak ve směru pohybu kočky, tak ve směru pohybu mostu. Řídicí PLC zpracovává povely od joysticku a zasílá požadované setpointy rychlosti do frekvenčního měniče, který ovládá motory a zajistí, že se jeřáb rozjede na požadovanou rychlost. Motory zrychlují a zpomalují zadaným konstantním zrychlením.

Požadavkem firmy bylo zajistit, aby se při manuálním řízení pomocí zmiňovaného joysticku břemeno rozkývalo minimálně. Požadavky na řízení byly finanční dostupnost a jednoduchost implementaci do současného PLC.

1.2 Navrhované řešení

Řešení, které splňuje požadavky, je přímovazební řízení. Nepotřebuje drahý senzor, který by snímal rozkmit břemene, je možné ho snadno implementovat do současného PLC a navíc je bezpečnější než zpětnovazební řízení. U zpětnovazebního řízení je totiž problém, že senzor snímající výchylku nebude přesný a navíc hrozí, že bude výstupní signál senzoru zašuměn. Tyto dva problémy mohou vézt k tomu, že systém může být spíše destabilizován než stabilizován.

Přímovazební řízení má za úkol tvarovat vstupní signál od joysticku tak, aby se minimalizovalo rozkývání břemene. V kapitole 2 je provedena rešerše jednotlivých metod přímovazebního řízení. V kapitole 3 je sestaven matematický model mostového jeřábu, který slouží pro testování jednotlivých metod. Kapitola 4 obsahuje testy jednotlivých řídících metod na modelu a výběr nejlepší metody, jež bude implementována do řídicího systému. V kapitole 5 je implementován a testován řídicí systém, který minimalizuje kyv břemene při dodržení bezpečnostních opatření a jednoduché obsluze operátorem.

2 Možnosti přímovazebního řízení

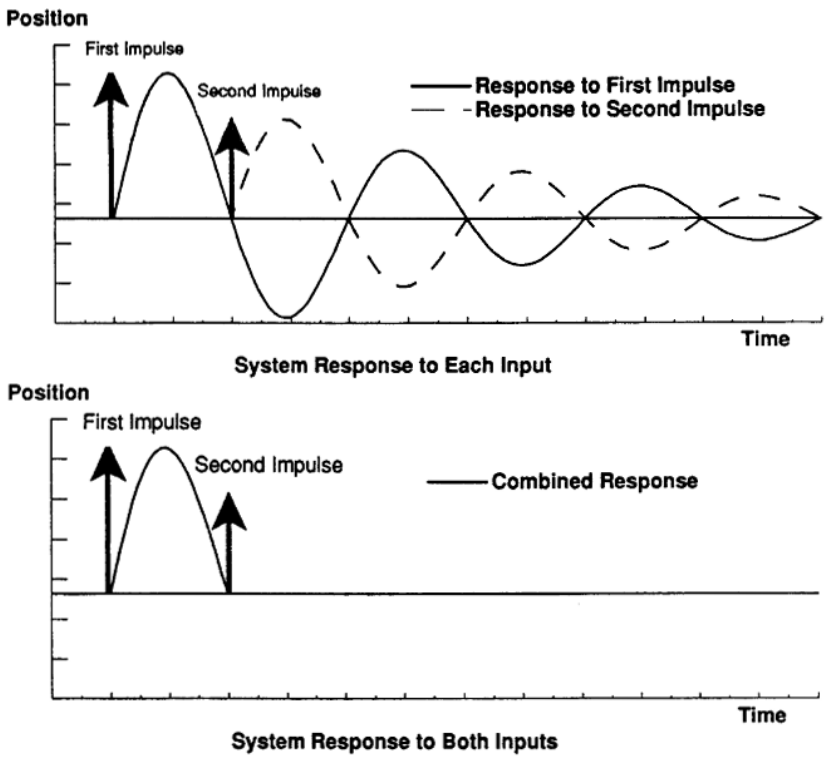
Tato kapitola se zabývá možnostmi přímovazebního řízení. Celkem jsem nalezl dvě metody, které v této kapitole popíši. Články z kterých jsem čerpal, jsou popsány v podkapitole 2.1. Obě metody přímovazebního řízení jsou založeny na tom, že se ve vstupním signálu potlačí frekvence, které mohou vybudit kmitavé módy řízené soustavy. Tyto metody jsou popsány v podkapitolách 2.2 a 2.3. Obě metody tedy tvarují průběh vstupního signálu. V případě systému firmy SIMO-CZ je vstupem rampa rychlosti, která vzniká podle zadaného setpointu rychlosti joystickem, jak je vidět např. na Obr. 3.2. Tato rampa rychlosti je tedy tvarovaná pomocí metod přímovazebního řízení.

2.1 Rešerše

Provedl jsem rešerši několika článků a prací, které se tímto tématem zabývají. Článek [1] popisuje možnosti přímovazebního řízení. Zmiňuje se o dvou metodách. První je input shaping a druhá je filtrování. Tyto metody navzájem porovnává, ale zkouší jenom několik input shaperů a pouze filtr typu dolní propust. Práce [2] je velmi obsáhlá a zmiňuje použití input shapingu zároveň se zpětnovazební regulací pomocí PID regulátoru. Input shaperů je zde zmíněno pět různých druhů. V práci je také ukázáno jaké mají tyto input shapery přenosy. Článek [3] je jedním ze starších článků, které se zabývají input shapingem. V tomto článku je input shaping odvozen, toto odvození je použito v kapitole 2.2.1. Článek také porovnává robustnost jednotlivých shaperů. Článek [4] porovnává navzájem input shaping a filtrování v různých testech, co se týče rychlosti náběhu systému a robustnosti. Článek [5] přistupuje k input shapingu přes vektorové diagramy, je zde dobře popsán jeden z nejrobustnějších shaperů tzv. EI-shaper.

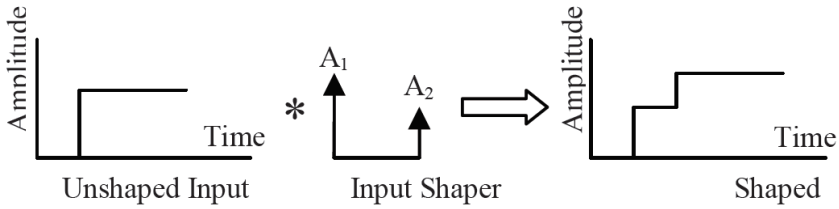
2.2 Input shaping

Input shaping je přímovazební technika určená pro zamezení oscilací výstupu přirozeně kmitajících systémů na určitých (známých) frekvencích. Myšlenka je jednoduchá, pomocí sekvence impulzů vstupního signálu o určité velikosti a vzájemné poloze v čase lze redukovat vibrace systému na minimum, jak je vidět na Obr. 2.1.



Obr. 2.1: Princip input shapingu [3]

Pomocí konvoluce série těchto impulsů a vstupního signálu je možné aplikovat input shaping na libovolný vstupní signál. Input shaping je vlastně filtr s impulsní charakteristikou jak ukazuje Obr. 2.2:



Obr. 2.2: Konvoluce shaperu a vstupního signálu [1]

Cenou za potlačení vibrací je nárůst doby náběhu[1] systému. Tato ztráta je ale bohatě nahrazena tím, že systém má daleko kratší dobu ustálení.

Drobnou nevýhodou je, že se na input shaping vztahují dva US patenty. První z roku 1990 číslo 4,916,635 je už neplatný. Druhý z roku 1997 číslo 5,638,267 je ještě v platnosti, ale jenom na území USA. Proto by neměl být problém s použitím input shapingu pro účely firmy SIMO-CZ.

[1] Doba náběhu je čas, ve kterém se změní odezva systému z 10% na 90% konečné hodnoty.

2.2.1 Odvození input shapingu pro lineární systém

Plné a detailnější odvození je ukázáno v [1], [3], zde je ukázán pouze nástin, jak tato metoda funguje.

Odezva tlumeného kmitajícího systému druhého řádu na vstupní impuls o určité amplitudě a času trvání limitně se blížícímu se nule je:

$$y(t) = \left[A \frac{\omega_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0(t-t_0)}\right] \sin(\omega_0\sqrt{1-\zeta^2}(t-t_0)) \tag{2.1}$$

Kde A je amplituda impulsu, ω_0 je přirozená frekvence kmitů systému, ζ je relativní tlumení systému, t je čas a t_0 je čas, kdy se objevil na vstupu daný impuls.

Jak je vidět, odezva systému je tlumený sinus. Superponováním odezev dvou vhodně zvolených vstupních impulsů lze docílit toho, že výsledná amplituda odezvy bude nulová. Názorně to je vidět na těchto matematických vzorcích:

$$B_1 \sin(\alpha t + \phi_1) + B_2 \sin(\alpha t + \phi_2) = A_{amp} \sin(\alpha t + \psi) \tag{2.2}$$

kde

$$\begin{gathered} A_{amp} = \sqrt{(B_1 \cos\phi_1 + B_2 \cos\phi_2)^2 + (B_1 \sin\phi_1 + B_2 \sin\phi_2)^2} \\ B_{1,2} = \frac{A_{1,2}\omega_0}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_0(t-t_{1,2})} \\ \alpha = \omega_0\sqrt{1-\zeta^2}t \\ \phi_{1,2} = -\omega_0\sqrt{1-\zeta^2}t_{1,2} \end{gathered} \tag{2.3}$$

a $t_{1,2}$ je čas, kdy se na vstupu objevil první, respektive druhý impuls.

Pro amplitudu tlumených harmonických kmitů A_{amp}, které vzniknou jako odezva systému na vstup o N impulsech, tedy platí:

$$A_{amp} = \sqrt{\left(\sum_{j=1}^{N} B_j \cos\phi_j\right)^2 + \left(\sum_{j=1}^{N} B_j \sin\phi_j\right)^2} \tag{2.4}$$

kde

$$\phi_j = \omega_0\sqrt{1-\zeta^2}t_j \qquad B_j = \frac{A_j\omega_0}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_0(t-t_j)} \tag{2.5}$$

A t_j je čas, kdy se na vstupu objevil *j-tý* impuls.

Aby systém nekmital po odeznění všech vstupních impulsů, musí platit, že rovnice (2.4) musí být rovna nule. Po dosazení rovnic (2.5) do rovnice (2.4) a splněním předchozí podmínky vychází:

$$\begin{aligned} &\sum_{j=1}^{N} A_j\, e^{-\zeta\omega_0(t-t_j)} \sin\left(t_j\omega_0\sqrt{1-\zeta^2}\right) = 0 \\ &\sum_{j=1}^{N} A_j\, e^{-\zeta\omega_0(t-t_j)} \cos\left(t_j\omega_0\sqrt{1-\zeta^2}\right) = 0 \end{aligned} \tag{2.6}$$

Člen $\frac{\omega_0}{\sqrt{1-\zeta^2}}$ v rovnicích (2.6) není, protože je konstantní a tím pádem mohl být vykrácen. Vyřešením obou rovnic pro N impulsů vzniká vstupní sekvence impulzů, která tvoří input shaper.

2.2.2 Druhy input-shapingu

Input shaperů existuje celá řada, zmíněny jsou např. v článku [2]. Každý je navržen jinou metodou a má specifické vlastnosti co se týče robustnosti a doby náběhu. V této práci uvedu tři shapery, které mi přišly nejpoužitelnější pro aplikaci na řízení mostového jeřábu.

2.2.2.1 ZV-shaper

Zero vibration shaper je technika, kde je shaper tvořen sérií dvou impulsů. Konvolucí těchto dvou impulsů a vstupního signálu vychází tvarovaný průběh. Impulsy musí splňovat podmínku (2.6) a navíc dvě podmínky [4]:

a) Impulsy musí být aplikovány pokud možno v co nejkratším čase, aby input-shaping nezpomaloval dobu náběhu.

b) Aby tvarovaný signál měl stejný setpoint jako původní, musí být splněna podmínka:

$$\sum_{j=1}^{N} A_j = 1 \tag{2.7}$$

Po splnění všech podmínek vypadají impulsy takto [4], [3]:

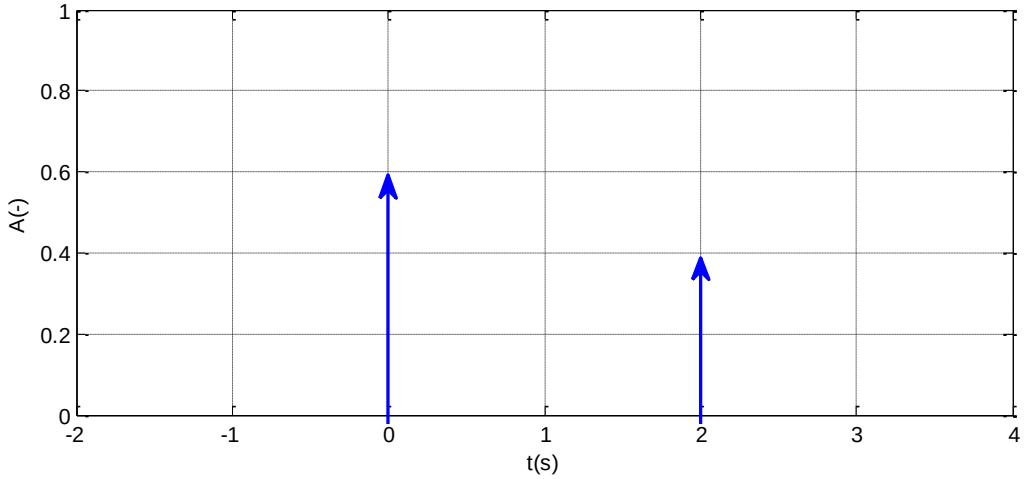
$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & 0{,}5T \end{bmatrix} \tag{2.8}$$

kde

$$T = \frac{2\pi}{\omega_0\sqrt{1-\zeta^2}} \qquad K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{2.9}$$

První řádek v matici rovnice (2.8) odpovídá amplitudě impulsů, druhý řádek této matice odpovídá poloze těchto impulsů v čase.

Pro názornost je na Obr. 2.3 vykreslena poloha a velikost impulsů pro $K = 0{,}67$ a $T = 4$:



Obr. 2.3: Poloha impulsů ZV-shaperu

Výhodou této metody je, že druhý impuls a zároveň poslední impuls je aplikován již v půlce periody kmitu. Nevýhodou je, že tento způsob je velmi citlivý na změnu vlastní frekvence kmitů. Robustnost[2] je dosažena při chybě přirozené frekvence maximálně $\pm 5\%$ [3].

2.2.2.2 ZVD-shaper

Zero vibration and derivative shaper je technika, kde musí být splněny podmínky jako pro ZV-shaper a navíc je přidána ještě jedna podmínka, která zvyšuje robustnost shaperu [4]:

$$\frac{d}{d\omega_0}\sqrt{\left(\sum_{j=1}^{N} B_j \cos\phi_j\right)^2 + \left(\sum_{j=1}^{N} B_j \sin\phi_j\right)^2} = 0 \tag{2.10}$$

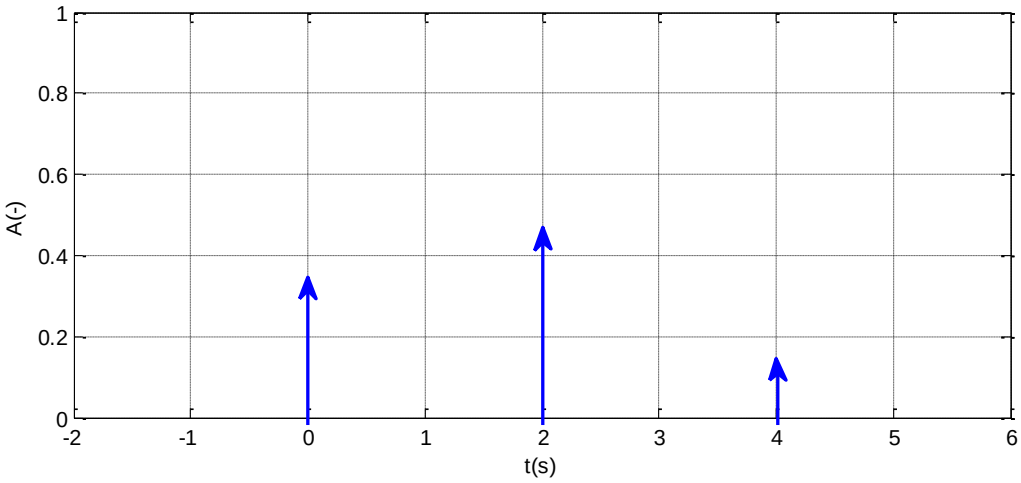
[2] Robustností je myšleno, že výstup systému bude oscilovat s amplitudou menší nebo rovnou 5% maximální amplitudě kmitů bez input shapingu.

Aby šly vyřešit všechny zmiňované podmínky, je potřeba přidat jeden impuls. Tímto se zvětší doba náběhu systému, na oplátku se ale zvětší robustnost shaperu. Vyřešením všech podmínek vychází pro sekvenci vstupních impulsů:

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+2K+K^2} & \frac{2K}{1+2K+K^2} & \frac{K^2}{1+2K+K^2} \\ 0 & 0{,}5T & T \end{bmatrix} \tag{2.11}$$

Podle rovnice (2.11) je vidět, že délka trvání sekvence je celá perioda kmitu, oproti ZV-shaperu (2.8), kde to byla jen polovina periody kmitu.

Pro lepší představu je vykreslen obrázek pro $K = 0{,}67$ a $T = 4$, tedy pro stejné hodnoty jako pro ZV-shaper:



Obr. 2.4: Poloha impulsů ZVD-shaperu

Obrovskou výhodou tohoto přístupu je zvýšení robustnosti shaperu, tj. shaper je robustní pro chybu v určení přirozené frekvence v rozmezí $\pm 20\%$.

2.2.2.3 EI-shaper

Extra-insensitive shaper je technika zvyšující robustnost. Tato metoda je nejrobustnější v porovnání s ostatními při zachování přijatelné rychlosti náběhu. Tento druh input-shapingu je založen na povolení malého zbytkového kývání V na vlastní frekvenci kyvu břemene (např. 5% z maximální amplitudy kyvu bez použití input shapingu). Díky tomu si ale dokáže poradit EI-shaper s velkou chybou v určení přirozené frekvence kmitů. Pro zbytkové kývání na úrovni $V = 5\%$ a tlumení $\zeta = 0{,}1$ vychází robustnost 47% pro chybné určení frekvence [5].

Vzhledem k nutnosti numerického řešení rovnic jsou vztahy (2.12) a (2.13) platné pro $0 \leq \zeta \leq 0{,}3$ a $0 \leq V \leq 0{,}15$, kde V je povolené zbytkové kývání na přirozené frekvenci [5]:

$$A_1 = 0{,}2497 + 0{,}2496V + 0{,}8001\zeta + 1{,}233V\zeta + 0{,}496\zeta^2 + 3{,}173V\zeta^2$$

$$A_2 = 1 - (A_1 + A_3) \tag{2.12}$$

$$A_3 = 0{,}2515 + 0{,}2147V - 0{,}8325\zeta + 1{,}415V\zeta + 0{,}8518\zeta^2 - 4{,}901V\zeta^2$$

$$T_1 = 0$$

$$T_2 = (0{,}5 + 0{,}4616V\zeta + 4{,}262V\zeta^2 + 1{,}756V\zeta^3 + 8{,}578V^2\zeta - 108{,}6V^2\zeta^2 + 337V^2\zeta^3)T \tag{2.13}$$

$$T_3 = T$$

EI-shaper má dobu náběhu stejnou jako ZVD-shaper, ale díky povolenému rozkmitu 5% z maximální amplitudy bez input shapingu na vlastní frekvenci kyvů břemene je robustnost EI-shaperu největší.

2.3 Filtrování

Pro filtrování je vhodné použít filtry typu dolní propust, nebo pásmová zádrž s tím, že se nepustí do systému signál na frekvenci rovné přirozené frekvenci kmitů, tím se zajistí, že se nevybudí kmitavé módy systému.

Filtry je možné navrhnout jako FIR nebo IIR, ale podle [4] tyto filtry prohrávají s input shapingem, hlavně co se týče složitosti, účinnosti potlačení zbytkového kývání a doby náběhu systému. V praktické části práce je tento fenomén potvrzen, viz kapitola 4.2.

3 Model mostového jeřábu

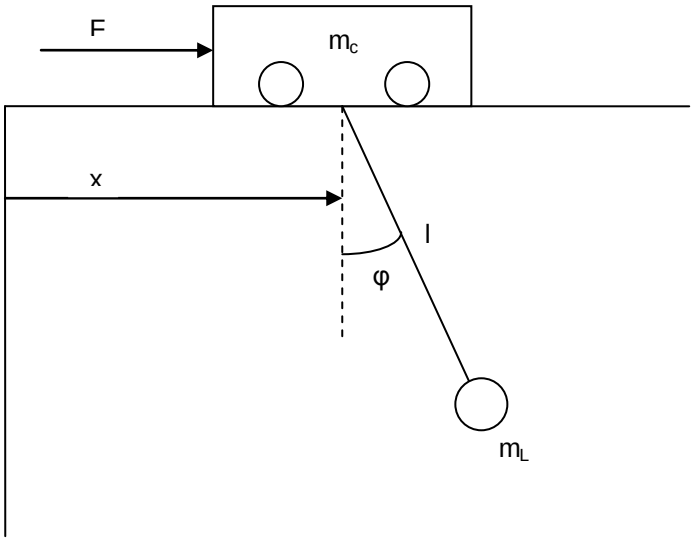
V této kapitole je pro potřebu testování řídicího systému vytvořen model, který musí dobře popisovat dynamiku jeřábu. K tomuto popisu jsem vybral diferenciální rovnice, které jsou odvozeny v publikacích [6], [7], [8]. V podkapitole 3.1.1 jsou tyto nelineární diferenciální rovnice linearizovány, protože parametry nutné pro výpočet přenosu input shaperu lze vyčíst pouze z lineárního modelu. V závěru kapitoly je model identifikován, aby odpovídal co nejvíce reálnému chování jeřábu.

Pro lepší vizualizace řízení je do modelu zakomponovaná virtuální realita, kde je možné jeřáb ovládat pomocí joysticku a ve 3D prostředí. Virtuální realita je převzata z [9].

3.1 Diferenciální rovnice soustavy

Mostový jeřáb lze modelovat jako kyvadlo zavěšené na vozíku, jak ukazuje Obr. 3.1, kde m_c je hmotnost kočky, m_L je hmotnost břemena, x je poloha vozíku, φ je úhel výkyvu břemena, l je délka lana a F je síla působící na kočku.

Takto popsaný model není úplný, protože popisuje dynamiku jen ve směru pojezdu kočky. Ve směru pojezdu mostu je ale situace téměř totožná, pouze m_c je nyní hmotnost celého mostu s kočkou. Celý model je tedy složen ze dvou sub-modelů. První popisuje dynamiku ve směru pohybu kočky a druhý ve směru pohybu mostu.



Obr. 3.1: Nákres modelu mostového jeřábu

Diferenciální rovnice popisující dynamiku systému ve směru pojezdu kočky:

$$m_L l\ddot{\varphi}_k + m_L g \sin\varphi_k = -m_L \ddot{x}_k \cos\varphi_k \tag{3.1}$$

$$(m_{ckocka} + m_L)\ddot{x}_k + m_L l\ddot{\varphi}_k \cos\varphi_k - m_L l\dot{\varphi}_k^2 \sin\varphi_k = F \tag{3.2}$$

Diferenciální rovnice popisující dynamiku systému ve směru pojezdu mostu:

$$m_L l\ddot{\varphi}_m + m_L g \sin\varphi_m = -m_L \ddot{x}_m \cos\varphi_m \tag{3.3}$$

$$(m_{cmost} + m_{ckocka} + m_L)\ddot{x}_m + m_L l\ddot{\varphi}_m \cos\varphi_m - m_L l\dot{\varphi}_m^2 \sin\varphi_m = F \tag{3.4}$$

kde g je tíhové zrychlení, m_{ckocka} je hmotnost kočky a m_{cmost} je hmotnost mostu. Ostatní proměnné již popisuje Obr. 3.1 s tím, že index k znamená, že se jedná o popis ve směru pohybu kočky a index m znamená, že jde o pohyb ve směru mostu.

Tyto rovnice uvažují jako vstup sílu F, z popisu skutečné konfigurace jeřábu SIMO-CZ (kap. 1.1.1) je ale zřejmé, že vstupem do soustavy je rychlost, kterou nastavují frekvenční měniče. Proto je možné soustavu rovnic (3.1) - (3.4) zjednodušit. Pro zjednodušení jsou zásadní rovnice, které popisují kyv břemene (3.1) a (3.3). Rovnice (3.2) a (3.4) popisují vliv kývajícího se břemene a vstupní síly na polohu mostu nebo kočky. To v případě reálného jeřábu SIMO-CZ ale není aktuální, jestliže je za vstup považována rychlost pohybu kočky nebo mostu. Frekvenční měnič společně s motorem působí jako zdroj rychlosti.

Zjednodušená rovnice (3.1) tedy vypadá:

$$m_L l\ddot{\varphi}_k + m_L g \sin\varphi_k = -m_L \dot{v}_k \cos\varphi_k \tag{3.5}$$

kde v_k je nastavená rychlost kočky frekvenčním měničem.

Než zjednoduším rovnice pro celý systém, je nutné si uvědomit, že v rovnicích (3.1) - (3.4) není uvažováno tlumení pro kývající se břemeno, to by ale neodpovídalo skutečnosti. Do rovnice (3.5) je tedy nutné vhodně zakomponovat tlumící člen:

$$m_L l\ddot{\varphi}_k + m_L g \sin\varphi_k = -m_L \dot{v}_k \cos\varphi_k - b_k l\dot{\varphi}_k \tag{3.6}$$

kde b_k je viskózní tlumící koeficient pro rovinu kyvu ve směru pohybu kočky. Síla vznikající viskózním tlumením je přímo úměrná rychlosti a je rovná $F_{dk} = b_k\dot{\varphi}_k$. Člen $b_k l\dot{\varphi}_k$ v rovnici (3.6) reprezentující právě tuto sílu, je navíc násoben délkou lana l, protože celá rovnice je tímto členem násobena. Po vydělení této rovnice délkou lana l už všechny členy rovnice reprezentují jednotlivé působící síly a tedy i síla F_{dk} je ve správném tvaru.

Takto je nutné přidat tlumící člen i do druhé rovnice (3.3) popisující kývání břemene ve směru pohybu mostu.

Dynamický model uvažovaného jeřábu lze tedy popsat rovnicemi:

$$\ddot{\varphi}_k = \frac{-g \sin \varphi_k - \dot{v}_k \cos \varphi_k}{l} - \frac{b_k}{m_L} \dot{\varphi}_k \tag{3.7}$$

$$\dot{x}_k = v_k \tag{3.8}$$

$$\ddot{\varphi}_m = \frac{-g \sin \varphi_m - \dot{v}_m \cos \varphi_m}{l} - \frac{b_m}{m_L} \dot{\varphi}_m \tag{3.9}$$

$$\dot{x}_m = v_m \tag{3.10}$$

Nelineární diferenciální rovnice (3.7) - (3.10) představují model systému, jehož vstupy jsou rychlost kočky v_k a rychlost mostu v_m. Výstupy jsou výchylka břemene v rovině kočky φ_k a v rovině mostu φ_m a poloha kočky x_k a mostu x_m.

3.1.1 Linearizace systému

Rovnice (3.7) - (3.10) simulující reálný systém jsou nelineární. Input shaping je však uvažován pro lineární systémy a tak je nutné rovnice linearizovat, aby z nich bylo možné vyčíst parametry potřebné pro shaper. Linearizace je oprávněná, protože výchylka břemene by neměla přesahovat $\varphi = 5°$. Linearizaci provedu kolem nulového úhlu výkyvu břemene, pak bude platit, že $\sin \varphi \sim \varphi$ a $\cos \varphi \sim 1$.

$$\ddot{\varphi}_k = \frac{-g\varphi_k - \dot{v}_k}{l} - \frac{b_k}{m_L} \dot{\varphi}_k \tag{3.11}$$

$$\dot{x}_k = v_k \tag{3.12}$$

$$\ddot{\varphi}_m = \frac{-g\varphi_m - \dot{v}_m}{l} - \frac{b_m}{m_L} \dot{\varphi}_m \tag{3.13}$$

$$\dot{x}_m = v_m \tag{3.14}$$

Rovnice (3.11) a (3.13) je nutné upravit do tvaru, aby z nich šly vyčíst parametry (2.9), čili perioda tlumených kmitů T a koeficient K, pomocí kterého se počítá velikost jednotlivých pulsů shaperu. Jak je vidět, tak rovnice (3.11) a (3.13) jsou druhého řádu a po úpravě by měly mít tento obecný tvar:

$$\ddot{\varphi} + 2\zeta\omega_0\dot{\varphi} + \omega_0^2\varphi = u \tag{3.15}$$

Úpravou rovnic (3.11) a (3.13) do tvaru jako má rovnice (3.15) vychází:

$$\ddot{\varphi}_k + \frac{b_k}{m_L}\dot{\varphi}_k + \frac{g}{l}\varphi_k = -\frac{1}{l}\dot{v}_k \tag{3.16}$$

$$\ddot{\varphi}_m + \frac{b_m}{m_L}\dot{\varphi}_m + \frac{g}{l}\varphi_m = -\frac{1}{l}\dot{v}_m \tag{3.17}$$

Porovnáním (3.16) a (3.17) s (3.15) vychází pro hledané parametry (2.9):

$$\zeta_k = \frac{b_k}{2m_L\omega_0} \qquad \zeta_m = \frac{b_m}{2m_L\omega_0} \qquad \omega_0 = \sqrt{\frac{g}{l}} \tag{3.18}$$

kde ζ_k a ζ_m je relativní tlumení kyvu břemene ve směru pohybu kočky resp. mostu v linearizovaném systému.

3.2 Identifikace modelu

Identifikovat budu pouze zjednodušený systém ovládaný rychlostí, protože takto to vypadá v realitě. Model potom sestavím v prostředí Simulinku a otestuji jeho chování.

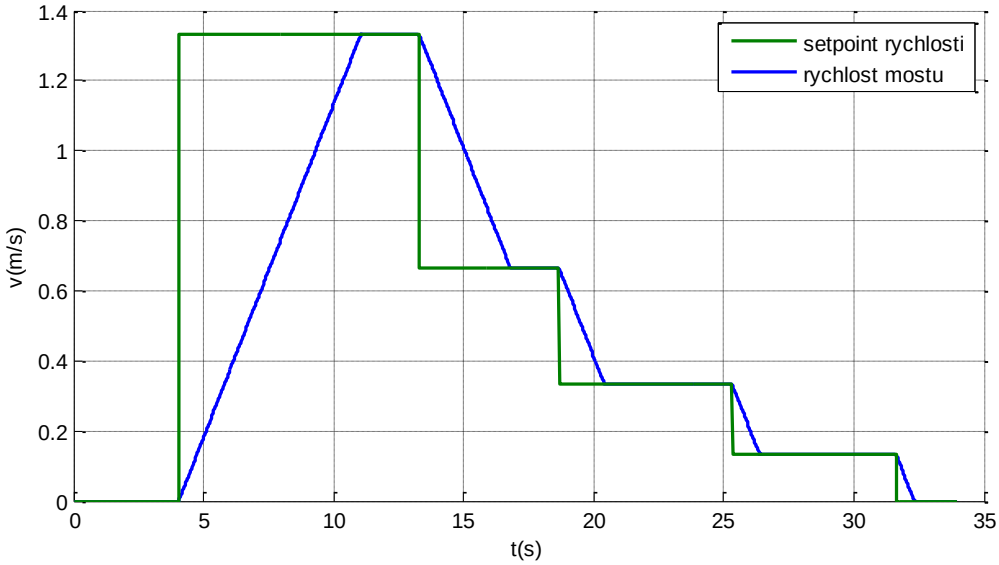
První neznámý parametr v rovnicích (3.7) - (3.10) je koeficient tlumení b_k a b_m. Tyto koeficienty jsem nastavil tak, aby kmity jeřábu nebyly příliš tlumené, protože to odpovídá realitě. Koeficienty jsem se snažil určit i podle videí s pohybem jeřábu, které byly dodány od firmy SIMO-CZ. Vzhledem k nízké kvalitě video záznamů je stanovení těchto koeficientů velmi nepřesné. Konkrétní hodnoty jsou $b_k = 120\ \frac{kg}{s}$ a $b_m = 120\frac{kg}{s}$.

Druhou neznámou veličinou v rovnicích je hmotnost břemene m_L, tzn. hmotnost drapáku a nákladu. Břemeno zadaného jeřábu může vážit v rozmezí hodnot $m_L = (1{,}5 - 6{,}3)t$. Tento parametr sice ovlivňuje relativní tlumení systému ζ, viz rovnice (3.18) , ale vzhledem k tomu, že je jeřáb málo tlumený systém, co se týče kmitů břemene, tak tento parametr příliš neovlivňuje výsledné chování systému. Pro zadané rozmezí hodnot parametru m_L a délku lana $l = 10m$ vychází relativní v rozmezí tlumení $\zeta = (0{,}04 - 0{,}01)$.

Třetí neznámou veličinou je délka lana l. Znalost délky lana je zásadní pro správnou funkci input shapingu. Délka lana totiž ovlivňuje periodu kmitů břemene. Tento parametr se na některých jeřábech měří a podle vyjádření firmy SIMO-CZ není problém toto měření na ostatních jeřábech dodělat. Pokud se délka měří, tak se měří pouze mezi kočkou a drapákem. Délku zavěšeného nákladu na drapáku zvětšující celkovou délku závěsu není možné změřit, tuto chybu musí eliminovat řídicí systém.

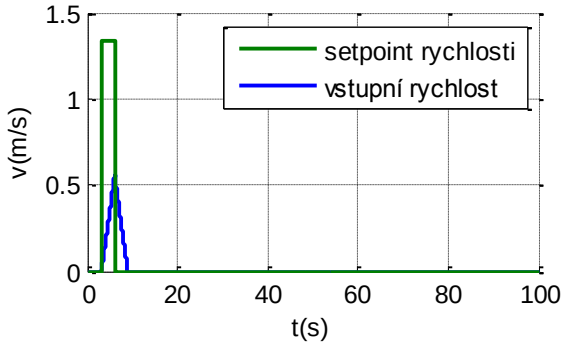
Vstupní rychlost v je hodně limitována díky omezeným možnostem frekvenčních měničů, které se chovají společně s motory jako zdroj rychlosti. Rychlost kočky může nabývat hodnot $v_k = (5 - 50)\frac{m}{min}$, rychlost mostu $v_m = (8 - 80)\frac{m}{min}$. Setpointu rychlosti dosahuje měnič tak, že rovnoměrně zrychluje kočkou $a_k = 0{,}119\ \frac{m}{s^2}$ nebo mostem $a_m = 0{,}190\ \frac{m}{s^2}$,

dokud se jeřáb nepohybuje požadovanou rychlostí. Rozjezd na maximální rychlost trvá jak u kočky, tak u mostu $t_{rozjezd} = 7s$. Brzdění z maximální rychlosti trvá stejnou dobu a brzdí se opět konstantním zrychlením. Na Obr. 3.2 je vidět pro lepší představu, jak funguje zpracování setpointu rychlosti, zadaného joystickem operátora, frekvenčním měničem. Na obrázku je zobrazen průběh rychlosti u mostu, u kočky to je analogicky stejné.

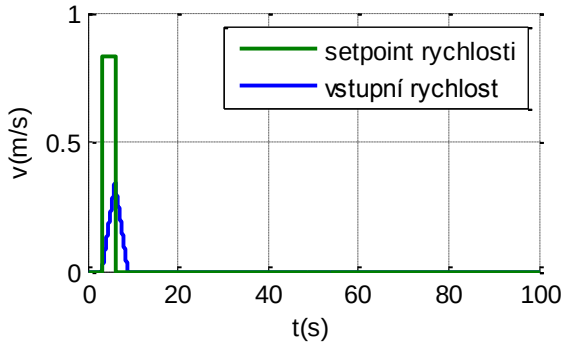


Obr. 3.2: Závislost rychlosti mostu na jejím setpointu

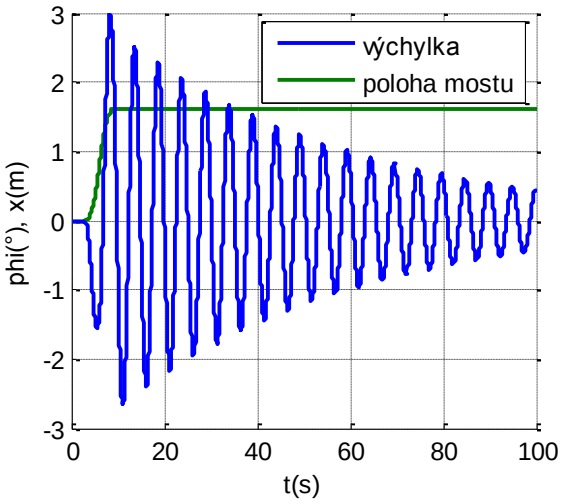
Nyní zbývá otestovat odezvu modelu pro kontrolu, jestli se model chová jako reálný systém. Na vstup systému přivedu trojúhelníkový puls rychlosti jak je vidět na obrázcích Obr. 3.3 a Obr. 3.4. Tento puls splňuje požadavky zadaného konstantního zrychlení:



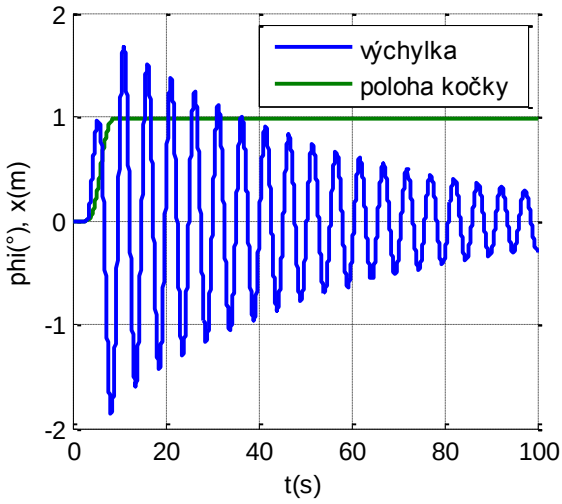
Obr. 3.3: Rychlost mostu



Obr. 3.4: Rychlost kočky



Obr. 3.5: Výchylka břemene ve směru pohybu mostu



Obr. 3.6: Výchylka břemene ve směru pohybu kočky

Jak je vidět z průběhů na obrázcích, tak se po vstupním krátkém trojúhelníkovém pulsu rychlosti systém rozkmitá tlumenými harmonickými kmity. Doba ustálení systému je celkem dlouhá v řádu minut. Vzhledem k tomu, že se špatně určuje správný koeficient tlumení, může se tato doba lišit od reálného jeřábu a v případě reálné aplikace, by se musela tato hodnota upravit.

Závěrem identifikace uvedu tabulky, které shrnují parametry a proměnné v modelu, kvůli snazší orientaci v jednotlivých hodnotách:

parametr	hodnota
g (tíhové zrychlení)	$9{,}8\ \frac{N}{kg}$
b_k (viskózní tlumení)	$120\ \frac{kg}{s}$
b_m (viskózní tlumení)	$120\ \frac{kg}{s}$

Tab. 3.1: Seznam parametrů a jejich hodnot

veličina	rozmezí
l (délka lana)	$(2{,}99 - 19)m$
m_L (hmotnost břemene)	$(1{,}5 - 6{,}3)t$
φ_k, φ_m (výchylka břemene)	$(\times)°$
v_k (rychlost kočky)	$(5 - 50)\frac{m}{min}$
v_m (rychlost mostu)	$(8 - 80)\frac{m}{min}$
x_k (poloha kočky)	$(0 - 20{,}1)m$
x_m (poloha mostu)	$(0 - 133)m$

Tab. 3.2: Seznam veličin a jejich hodnot

Kde $(\times)°$ znamená, že hodnota není nijak omezená. V případě výchylky břemene je hodnota omezena, ale až sekundárně jako výsledek navrhnutého přímovazebního řízení, proto uvádím, že hodnota není omezena. Řídicí systém má tedy úkol minimalizovat výchylku břemene a tím ji vlastně omezit do určitých mezí.

4 Aplikace řízení na modelu

V této části jsou otestovány řídicí metody na vytvořeném modelu jeřábu. Nejdříve je otestován input shaping a všechny druhy uvedených shaperů, poté se testuje filtrování. V závěru této kapitoly je provedeno srovnání, která ze dvou metod, ať input shaping nebo filtrování, je lepší.

Testování provedu tak, že nechám jeřáb rozjet na maximální rychlost a po několika vteřinách začnu brzdit, dokud se jeřáb nezastaví. Přitom budu sledovat odezvu systému pouze pro rovinu pohybu mostu, v rovině pohybu kočky vypadají testy podobně.

4.1 Input Shaping

Pro možnosti odzkoušení input shapingu je nutné převést shaper do podoby, kterou lze implementovat do modelu. Nabízí se dvě možnosti. Buď se shaper navrhne jako spojitý filtr. Poloha impulsů je určena jako dopravní zpoždění, viz (4.1). Druhá možnost je navrhnout shaper jako diskrétní člen. S tím že poloha impulsů se reprezentuje také jako zpoždění, zde ale v záporných mocninách diskrétního operátoru z, viz (4.2).

Spojitá verze (nejjednodušší shaper: ZV-shaper) [2]:

$$ZV(s) = A_1 + A_2 e^{-s\frac{T}{2}} \tag{4.1}$$

Diskrétní verze:

$$ZV(z) = A_1 + A_2 z^{-round\left(\frac{T}{2T_s}\right)} \tag{4.2}$$

kde A_1, A_2 jsou amplitudy impulsů, T je perioda vlastních kmitů břemene, T_s je vzorkovací perioda, s je Laplaceův operátor a z je operátor z-transformace. Operace *round* znamená, že se proměnná uvnitř funkce zaokrouhlí, protože mocniny operátoru z mohou být pouze celá čísla.

Vzhledem k tomu, že do reálného řídicího systému implementovaného v PLC půjde naprogramovat spíš diskrétní verze přenosu, než spojitá, budu provádět testy na diskrétní verzi shaperu. Periodu vzorkování zvolím $T_s = 0{,}03s$. PLC sice zvládne vzorkovat rychleji, když se započítají zpoždění vstupů a přenosy po sběrnici, vychází perioda vzorkování přibližně $T_{smin} = 0{,}01s$. Podle simulačních testů ale $T_s = 0{,}03s$ dostačuje.

Všechny testy se budou provádět vždy pro shaper s přesně definovaným přenosem. V reálném systému bude ale input shaper implementován tak, že při změně délky závěsu, hmotnosti břemene nebo periody vzorkování se přenos vždy přepočítá podle aktuálních parametrů.

4.1.1 ZV-shaper

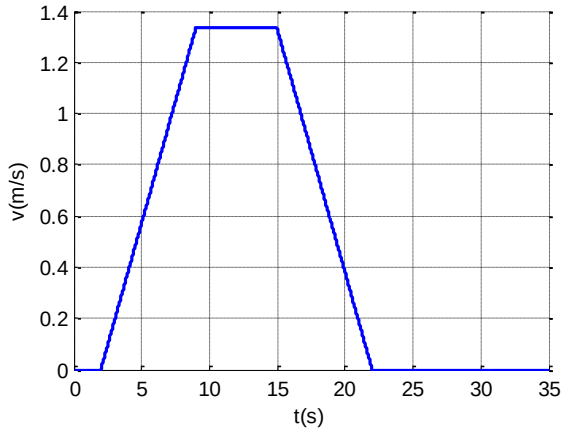
ZV-shaper je vůbec nejjednodušší metoda pro input shaping. Používá sérii dvou impulzů, jejich vlastnosti jsou vidět v (2.8). Přenos takovéto shaperu v diskrétní oblasti je už zmíněn výše v (4.2).

Pro konkrétní hodnoty parametrů, délka lana $l = 5m$, hmotnost břemene $m_L = 3t$, koeficientu tlumení $b_m = 120\frac{kg}{s}$ a periodě vzorkování $T_s = 0{,}03s$, vychází přenos ZV-shaperu:

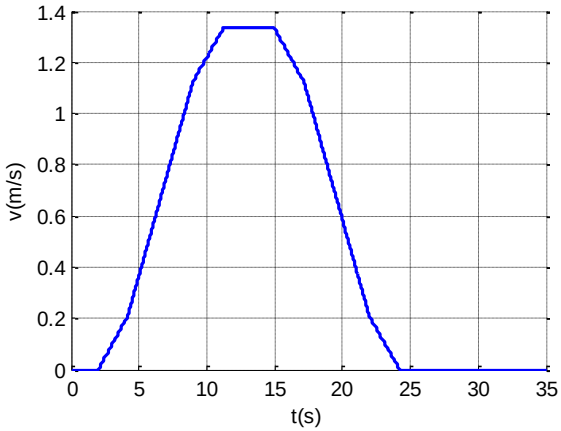
$$ZV(z) = \frac{0{,}5101z^{67} + 0{,}4899}{z^{67}} \tag{4.3}$$

Tento vzniklý přenos použiji jako input shaper a provedu testy řízení na modelu.

První test provedu za předpokladu, že input shaper zná přesně všechny parametry systému:

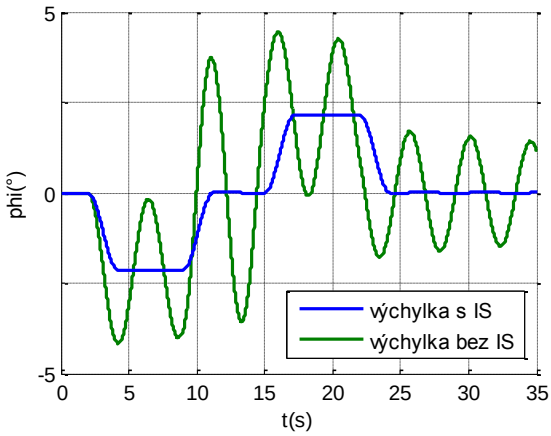


Obr. 4.1: Požadovaná vstupní rychlost

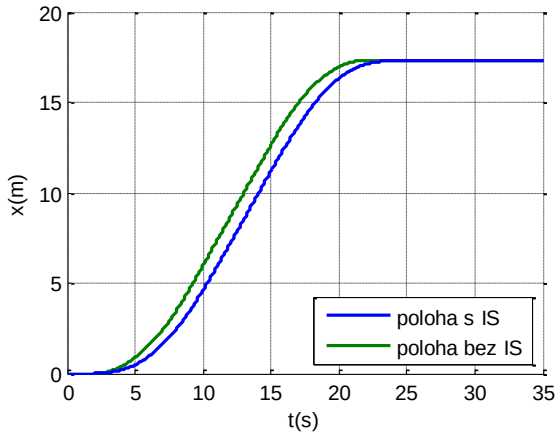


Obr. 4.2: Rychlost po průchodu shaperem

Nyní už by měla rychlost splňovat podmínku, že neobsahuje energii na frekvenci, která se rovná vlastní frekvenci kmitů břemene, a tudíž by se nemělo břemeno rozkývat:



Obr. 4.3: Porovnání výchylky břemene

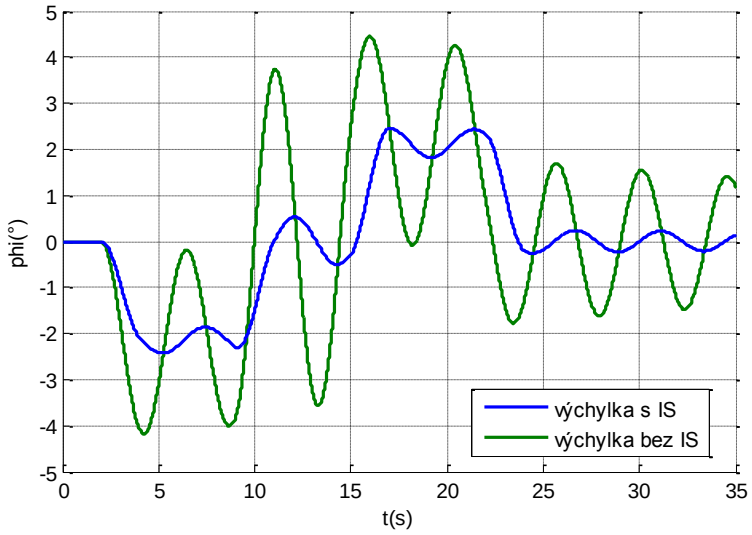


Obr. 4.4: Porovnání rychlosti náběhu systému

Jak je vidět na Obr. 4.3, pokud ZV-shaper zná všechny parametry přesně, tak dokáže zajistit nulovou výchylku břemene nejen po dojetí jeřábu do koncové polohy, ale i v ustálené rychlosti. Na druhou stranu je zřejmé z Obr. 4.4, že je systém s input shapingem o něco pomalejší než systém bez input shapingu. Přesto je input shaping výhodný. Ačkoli trvá déle jeřábu, než se dostane na požadovanou polohu, bude mít v tomto místě nulovou výchylku břemene na rozdíl od systému bez input shapingu, kde bude muset operátor svými schopnostmi omezit kývání jeřábu, což mu bude ve výsledku trvat déle.

Nevýhodou jednoduchosti ZV-shaperu je jeho velká náchylnost ke změně parametrů. Jak už je zmíněno v kapitole 2.2.2.1, tak ZV-shaper je robustní vůči změně vlastní frekvence o $\pm 5\%$.

Robustnost otestuji zavedením 10% chyby do určení vlastní frekvence kmitů břemene:



Obr. 4.5: Robustnost ZV-shaperu

Jak je vidět na Obr. 4.5, tak výchylka břemene systému s input shapingem není ještě tak markantní oproti neřízenému systému, přesto už nesplňuje podmínku robustnosti. Výchylka má totiž velikost 14% maximální výchylky neřízeného systému, oproti 5% povoleným.

4.1.2 ZVD-shaper

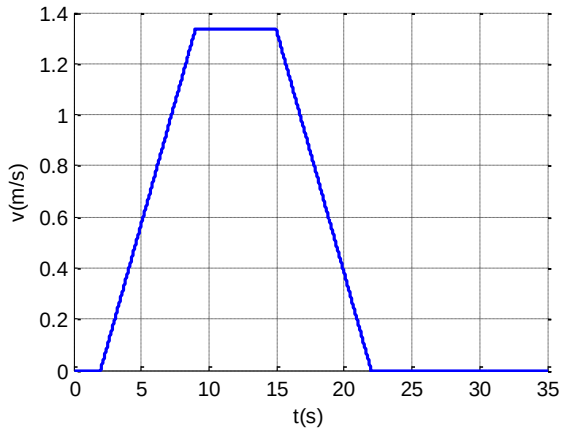
ZVD-shaper je robustnější varianta input shapingu. Tato metoda využívá sekvenci tří pulsů namísto dvou. Splněním podmínky (2.10) se dosahuje vyšší robustnosti v porovnání s jednoduchým ZV-shaperem. Diskrétní přenos shaperu je:

$$ZVD(z) = A_1 + A_2 z^{-round\left(\frac{T}{2T_s}\right)} + A_3 z^{-round\left(\frac{T}{T_s}\right)} \tag{4.4}$$

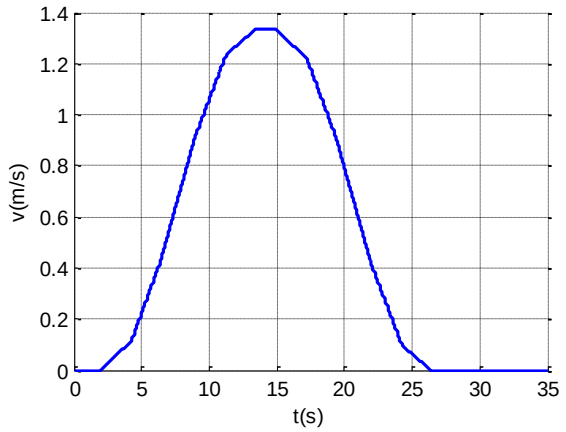
Pro konkrétní hodnoty parametrů jako pro ZV-shaper v kapitole 4.1.1 vychází přenos ZVD-shaperu:

$$ZVD(z) = \frac{0{,}2607z^{135} + 0{,}4998z^{68} + 0{,}24}{z^{135}} \tag{4.5}$$

První test provedu stejně jako u ZV-shaperu. Nejprve budu předpokládat, že shaper zná všechny parametry přesně:



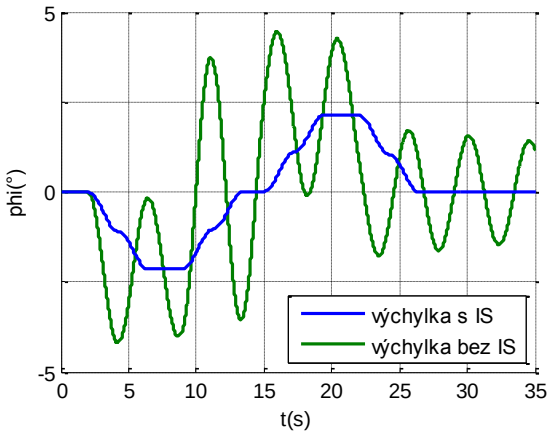
Obr. 4.6: Požadovaná rychlost



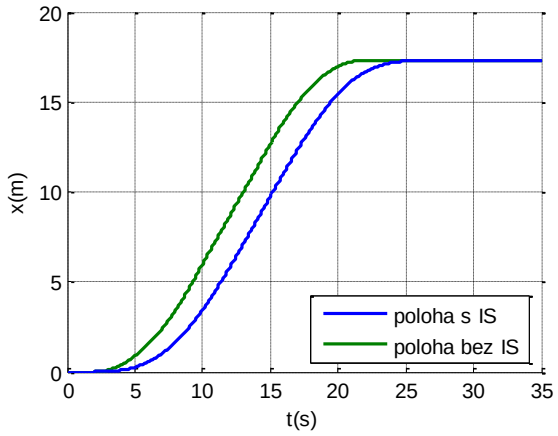
Obr. 4.7: Rychlost po průchodu shaperem

Jak je vidět na obrázcích, tak shaper opět natvaroval vstupní rychlost, ale tentokrát na rozdíl od Obr. 4.2 je na Obr. 4.7 vidět, že je vstup více tvarován, vypadá více zaobleně. To je právě způsobeno tím, že vstup tvarují tři impulzy místo dvou. Také je vidět, že natvarování vstupu trvá delší dobu než v případě Obr. 4.2.

Na Obr. 4.8 a Obr. 4.9 je vidět odezva systému:

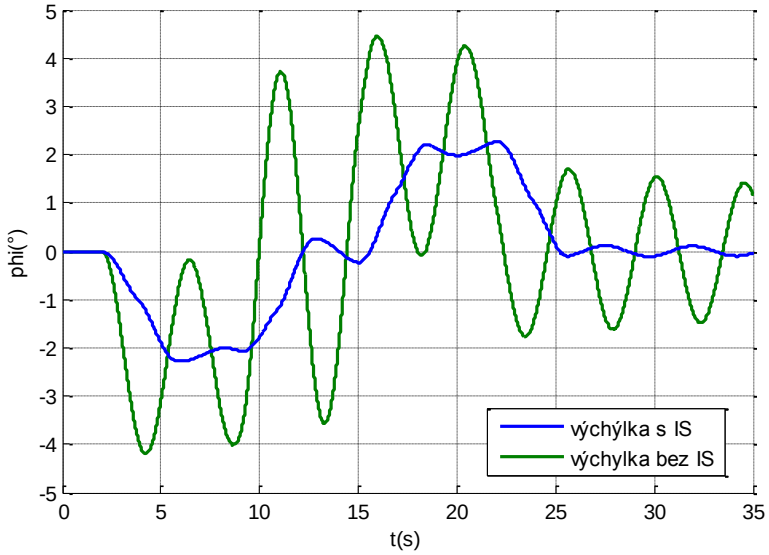


Obr. 4.8: Porovnání výchylky břemene



Obr. 4.9: Porovnání rychlosti náběhu systému

Jak je vidět na Obr. 4.9, tak je systém pomalejší než při použití ZV-shaperu, viz Obr. 4.4. Tento jev může být pro operátora trochu nepříjemný a musí s ním počítat, jeřáb se rozjíždí a brzdí pomaleji než, když se použije neřízený systém. Na oplátku tato metoda poskytuje velkou robustnost proti změně parametrů. Proto robustnost otestuji a zavedu do systému 20% chybu v určení vlastní frekvence kmitů břemene:



Obr. 4.10: Robustnost ZVD-shaperu

Jak je vidět na Obr. 4.10, tak se břemeno skutečně rozkýve málo, i když je do systému zavedená celkem velká chyba. Podmínka robustnosti je tedy splněna pro chybu v určení parametrů $\pm 20\%$.

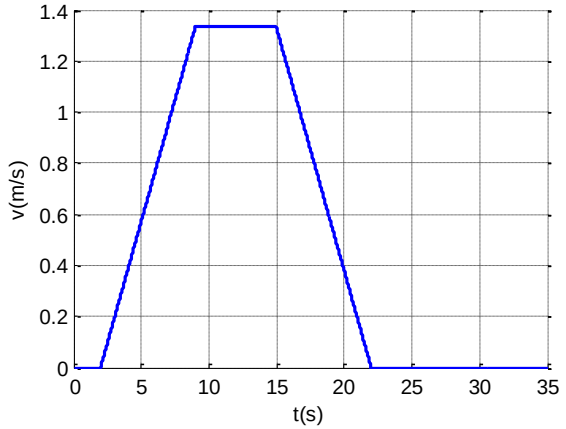
4.1.3 EI-shaper

Jedním z nejvíce robustních, při zachování stejné rychlosti náběhu systému jako u ZVD-shaperu, je právě EI-shaper. Využívá podobně jako ZVD-shaper tří impulsů pro tvarování vstupního signálu. Dokáže je ale tak vhodně umístit tak, že robustnost tohoto input shapingu je ±30%, pro poměrně malé tlumení. Pokud je systém více zatlumen, dosahuje EI-shaper ještě větší robustnosti. Zvýšení robustnosti je dosaženo tím, že se povolí zbytkové kývání (v procentech maximální výchylky břemene systému bez input shapingu na vlastní frekvenci kmitů) při přesné znalosti všech parametrů.

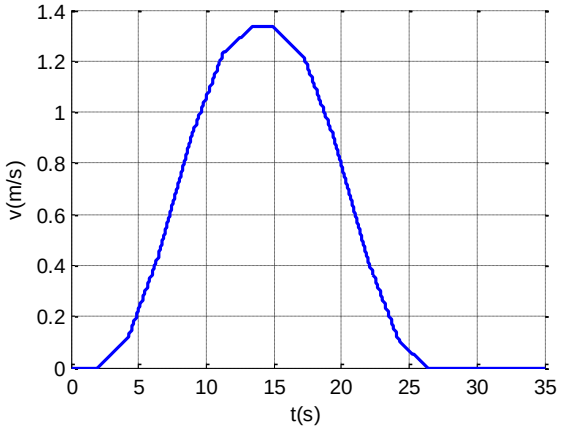
Pro konkrétní hodnoty parametrů jako pro ZV-shaper v kapitole 4.1.1 a pro zbytkové kývání $V = 5\%$ vychází diskrétní přenos EI-shaperu:

$$EI(z) = \frac{0{,}2724z^{123} + 0{,}4742z^{61} + 0{,}2533}{z^{123}} \tag{4.6}$$

U prvního testování se opět předpokládá, že shaper zná všechny parametry přesně:



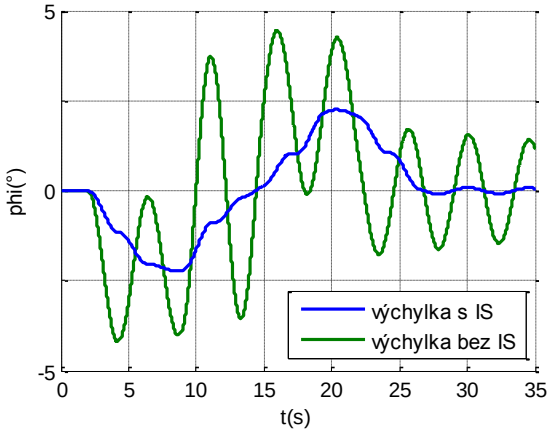
Obr. 4.11: Požadovaná rychlost



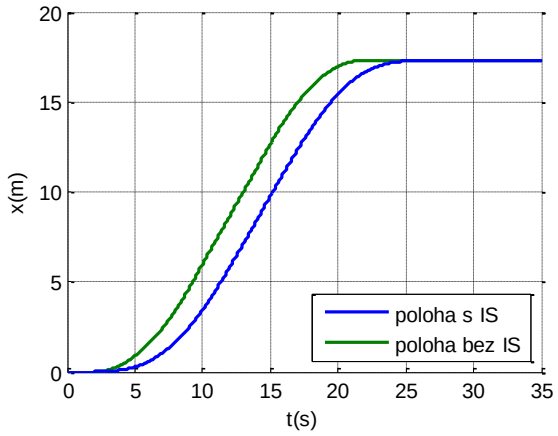
Obr. 4.12: Rychlost po průchodu shaperem

Vytvarovaná rychlost Obr. 4.12 vypadá podobně jako u ZVD-shaperu Obr. 4.7. Je to způsobeno tím, že oba shapery používají ke tvarování tři impulsy a jejich výstup bude tedy vypadat podobně.

Odezva systému tedy vypadá:

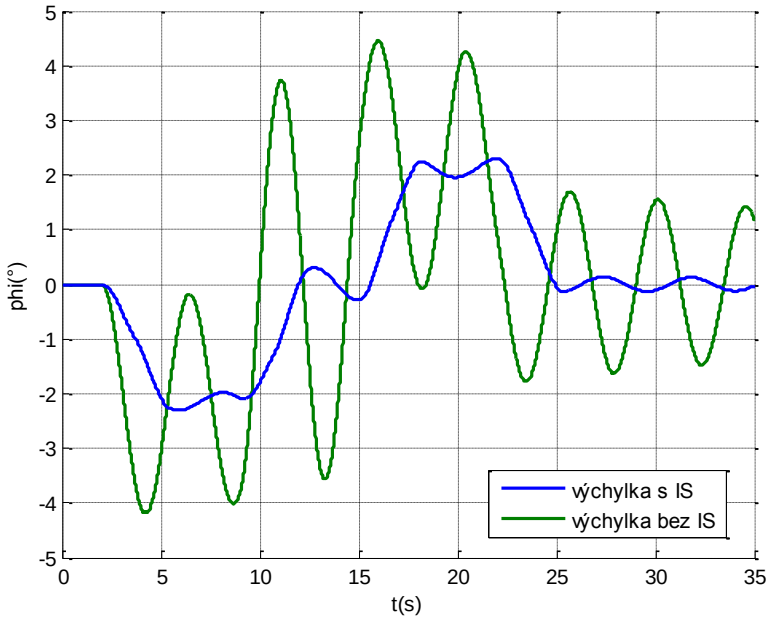


Obr. 4.13: Porovnání výchylky břemene



Obr. 4.14: Porovnání rychlosti náběhu systému

Tato metoda způsobuje podobnou rychlost náběhu systému jako ZVD-shaper. Výhodou oproti ZVD-shaperu je zvýšení robustnosti, díky povolenému zbytkovému kývání na vlastní frekvenci kmitů. Pro test zavedu 30% chybu do systému v chybném určení přirozené frekvence:



Obr. 4.15: Robustnost EI-shaperu

Jak je vidět na Obr. 4.15, tak se břemeno kývá jenom málo a udávaná robustnost $\pm 30\%$ platí.

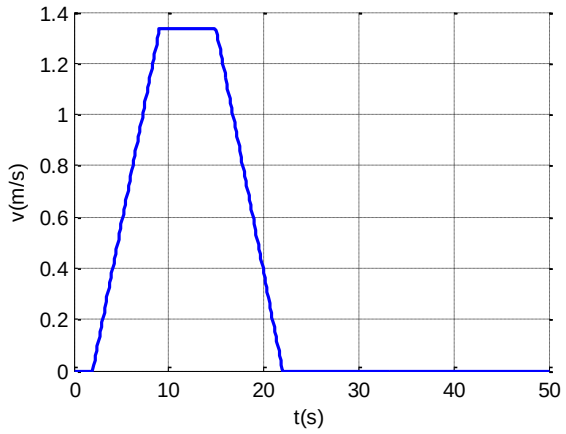
4.2 Filtrování

Pro filtrování otestuji diskrétní FIR filtry typu dolní propust a pásmová zádrž.

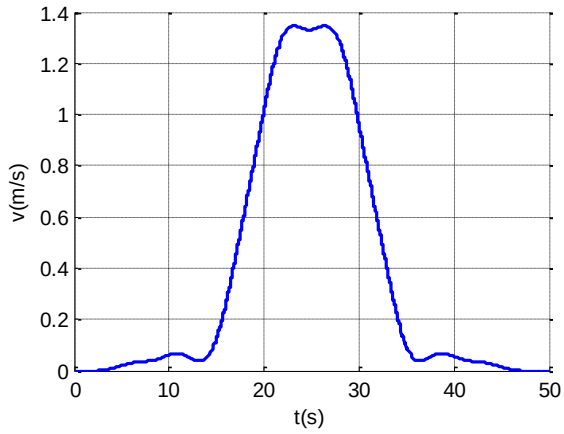
4.2.1 FIR – dolní propust

Tento filtr se pokusím navrhnout jako v [4]. Filtr má za úkol odfiltrovat frekvence, na kterých se pak nedostává energie, která by vybudila kmitavé módy. Bohužel se tímto filtrem oříznou i vyšší frekvence, které by mohly urychlit odezvu systému. K testování použiji FIR filtr s parametry $\omega_{prop} = 0{,}8\omega_v$ a $\omega_{zadr} = \omega_v$, kde $\omega_v = \sqrt{1 - \zeta_{linm}^2}$. Řád filtru je $N = 256$:

Výsledky testování:



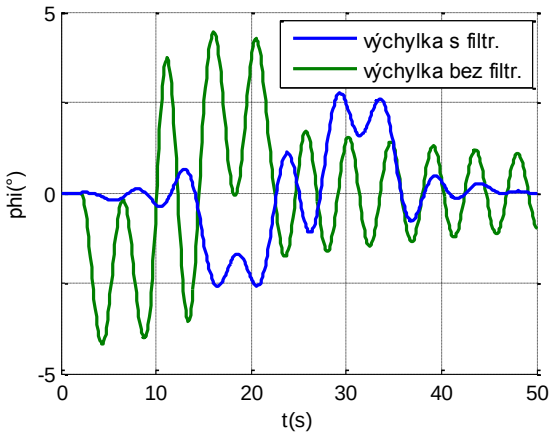
Obr. 4.16: Požadovaná rychlost



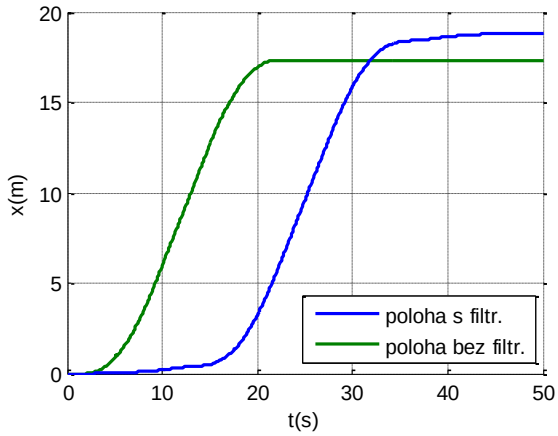
Obr. 4.17: Rychlost po průchodu filtrem

Jak je vidět na Obr. 4.17 tak filtr vytvaruje rychlost, ale je zde vidět, že má značné zpoždění oproti vstupnímu signálu.

Odezva systému je vidět na Obr. 4.18 a na Obr. 4.19:



Obr. 4.18: Porovnání výchylky břemene



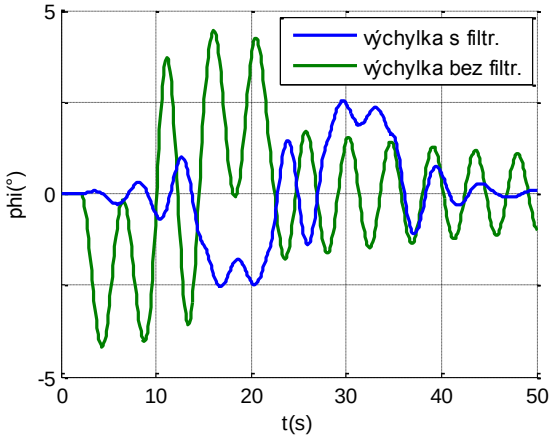
Obr. 4.19: Porovnání rychlosti systému

Nejenže zavádí filtr do systému veliké zpoždění, ale ještě je vidět na Obr. 4.19, že most přejede místo, kde neřízený jeřáb normálně zastaví. Je to způsobeno tím, že filtr nedodrží v propustném pásmu přesně jednotkové zesílení.

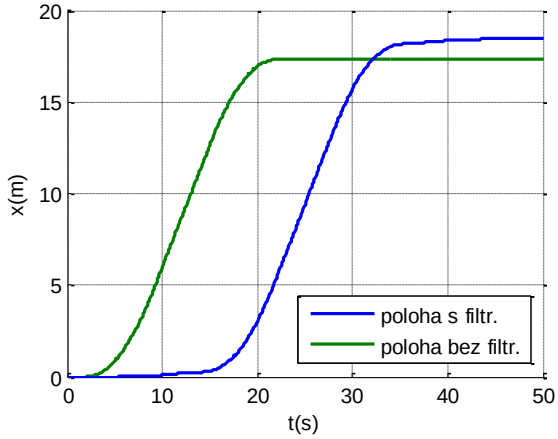
4.2.2 FIR – pásmová zádrž

Vhodnější filtr než dolní propust by měla být pásmová zádrž, která odřízne pouze ty frekvence, které by mohly způsobit rozkmitání břemene. Parametry filtru jsem zvolil podle [4]: $\omega_{prop1} = 0{,}8\omega_v$, $\omega_{stop1} = 0{,}95\omega_v$, $\omega_{stop2} = 1{,}05\omega_v$ a $\omega_{prop2} = 1{,}2\omega_v$. Filtr je typu FIR řádu $N = 256$.

Vzhledem k tomu, že filtry mají velké zpoždění, vykreslím jenom průběh výchylky břemene a polohy mostu. Průběhy rychlostí jsou podobné jako na Obr. 4.17:



Obr. 4.20: Porovnání výchylky břemene



Obr. 4.21: Porovnání rychlosti systému

Jak je vidět, tak má filtr podobný průběh a podobné zpoždění jako dolní propust, velkou výhodu tedy pásmová zádrž nepřináší.

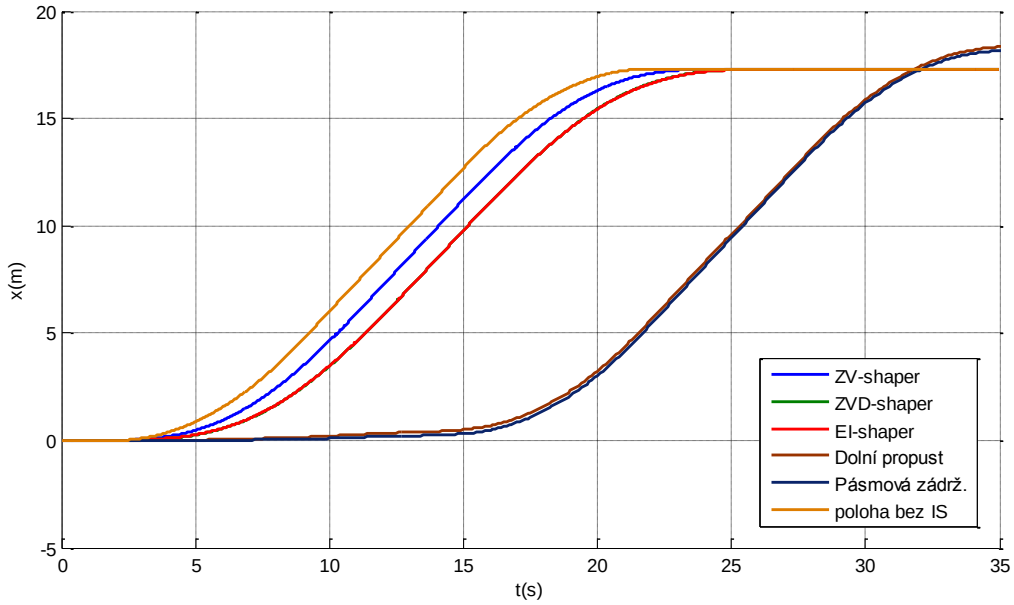
4.2.3 IIR filtry

Tyto filtry sice budou rychlejší než testované FIR filtry, ale podle článku [4] je jejich odezva stále několikrát pomalejší než odezva input shaperů. Navíc je nevýhodné, že IIR filtry například podle [10] obsahují na rozdíl od FIR filtrů póly mimo počátek komplexní roviny a tudíž se mohou dostat za hranice jednotkového kruhu a tím pádem může být IIR filtr, na rozdíl od FIR filtru nestabilní. Kvůli zmíněným vlastnostem nebudu IIR filtry testovat.

4.3 Porovnání obou metod řízení

Jedním z hlavních kritérií při porovnávání je rychlost náběhu systému, při zajištění minimálního kyvu břemene. Operátorovi se totiž bude velmi špatně řídit jeřáb, který nebude několik vteřin reagovat, než se začne rozjíždět, popřípadě brzdit, právě proto je velmi důležitá rychlost náběhu.

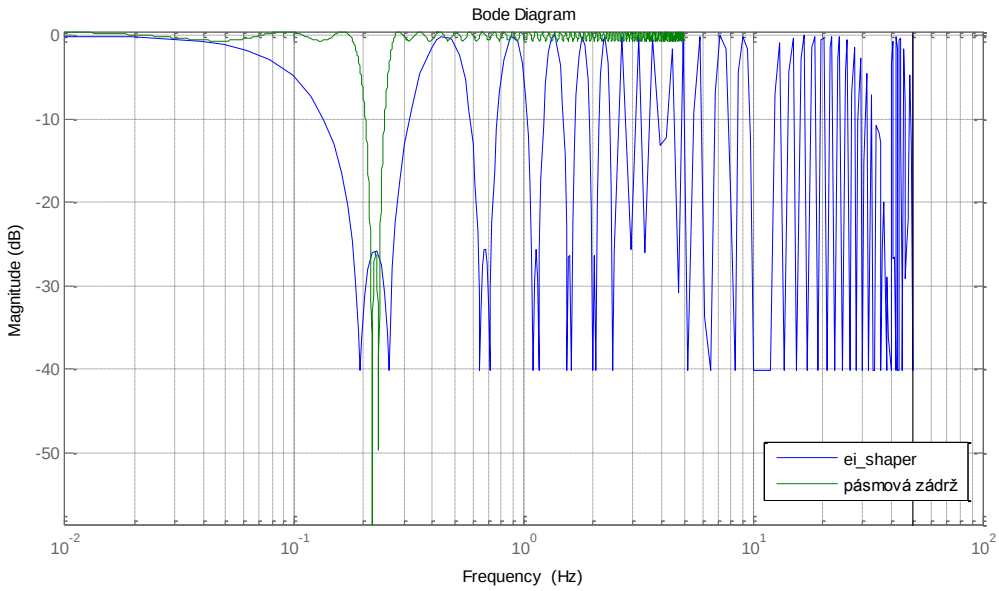
Porovnání rychlostí náběhu všech řídících metod zmiňovaných v kapitolách 4.1 a 4.2:



Obr. 4.22: Srovnání rychlosti náběhu systému při použití všech navrhovaných metod řízení

Jak je vidět na Obr. 4.22, tak nejlepších výsledků dosahuje ZV-shaper, o trochu pomalejší jsou na stejném místě ZVD-shaper a EI-shaper. Daleko za všemi ostatními zůstávají klasické filtrovací metody.

Jako další srovnání je možné srovnat input shaping a filtrování ve frekvenční oblasti, vykreslím Bodeho charakteristiku pro diskrétní přenos EI-shaperu a FIR filtru – pásmové zádrže:



Obr. 4.23: Porovnání Bodeho charakteristiky EI-shaperu a FIR filtru – pásmové zádrže

Pro porovnání jsem nechal vykreslit jenom amplitudovou část. Fázová charakteristika není potřeba, protože zpoždění jednotlivých metod je porovnáno v testu rychlosti náběhu řízeného systému. Jak je vidět na Obr. 4.23, tak se EI-shaper chová podobně jako pásmová zádrž. Na obrázku je také vidět, že zádržné pásmo je u EI-shaperu daleko širší, což právě způsobuje tak velkou robustnost.

Jak test porovnávající rychlost náběhu řízeného systému při použití všech vyzkoušených metod, tak test srovnávající amplitudovou frekvenční charakteristiku FIR filtru – pásmová zádrž a EI-shaperu prokázal, že je pro řízení vhodnější použít input shaping. Do reálné aplikace se tedy bude implementovat tato metoda. Z input shaperů vyberu EI-shaper, díky jeho velké robustnosti.

5 Aplikace řízení na reálném jeřábu

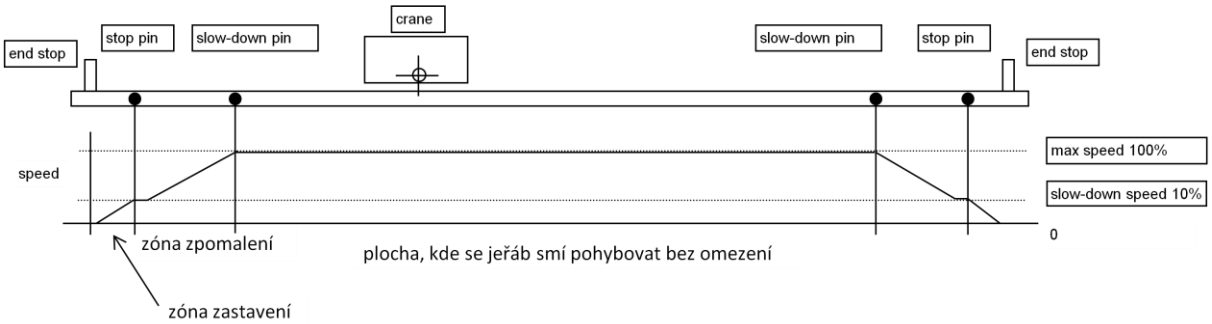
V této kapitole se pokusím navrhnout způsob, jak by se mohl naimplementovat input shaping do stávajícího řídicího systému. Jeřáb je v současné době řízen pomocí PLC, takže je nutné implementovat řízení pokud možno jednoduše. Výhodně jsem k tomu využil embedded matlab function v Simulinku. Ta také podporuje pouze základní operace jako jsou aritmetické operace, popřípadě jednoduchou rozhodovací logiku. Proto budu řídicí systém implementovat právě v tomto kódu, který je je relativně jednoduchý a měl by být dobře srozumitelný pro programátora, jež bude implementovat tento systém do PLC. Další výhodou je možnost otestování na modelu, zda můj navrhnutý algoritmus funguje správně a zdali splňuje bezpečnostní opatření, viz podkapitola 5.1.

5.1 Bezpečnostní opatření

Řídicí algoritmus musí zajistit, že při jakkoli neopatrné manipulaci s jeřábem, zůstane jeřáb v oblasti, která mu je vymezena. Musí být zajištěno, že operátor nemůže svým počínáním vyjet mimo povolenou oblast.

V současnosti jsou do systému zapojeny křížové přepínače, které zajišťují, bezpečnost ve dvou úrovních. Rozdělují plochu, kde se může jeřáb pohybovat do tří zón – zóna bez omezení rychlosti, zóna zpomalení a zóna zastavení. Při přejezdu jeřábu z plochy, kde se může volně pohybovat do zóny zpomalení, platí, že jeřáb musí stihnout zpomalit z jakékoli rychlosti na deset procent maximální rychlosti. Tato zóna hraničí se zónou zastavení, která se nachází úplně po okrajích povolené plochy pro pohyb jeřábu. Tato zóna zajišťuje, že pokud do ní jeřáb vjede, tak zastaví z oněch deseti procent maximální rychlosti, kterou se může pohybovat v předchozí zóně – zóně zpomalení. Nesmí se stát, že by jeřáb z této zóny vyjel, protože by mohlo dojít ke kolizi se stěnami anebo předměty v okolí.

Graficky tuto problematiku popisuje Obr. 5.1:



Obr. 5.1: Bezpečnostní zóny

Samotný křížový přepínač je na Obr. 5.2:

Obr. 5.2: Křížový přepínač

Funkce křížového přepínače, který je na Obr. 5.2, je následující. V místech kde jeřáb najíždí do zóny zpomalení, či zastavení je umístěna tyčka. Křížový přepínač je umístěn na jeřábu. V okamžiku kdy se jeřáb dostane do zóny, křížový přepínač narazí do tyčky a ta ho otočí o jednu obrátku. Takto PLC detekuje, v jaké zóně se jeřáb nachází.

Dalším požadavkem je, že když se jeřáb rozjíždí ze zón zpomalení nebo zastavení směrem do plochy, kde se může volně pohybovat, rozjíždí se jako by v žádné zóně nebyl, tedy klidně i maximální rychlostí.

V současné době mají obě zóny rozměry:

- Šířka zastavovací zóny ve směru pohybu kočky je 0,7m od kraje
- Šířka zastavovací zóny ve směru pohybu mostu je 1,5m od kraje
- Šířka zpomalovací zóny ve směru pohybu kočky je 2m od zastavovací zóny
- Šířka zpomalovací zóny ve směru pohybu mostu je 5m od zastavovací zóny

Vzhledem k tomu, že systém s input shapingem má pomalejší dobu náběhu, platí, že bude brzdit déle než systém bez input shapingu. Proto se rozměry jednotlivých zón musí opravit:

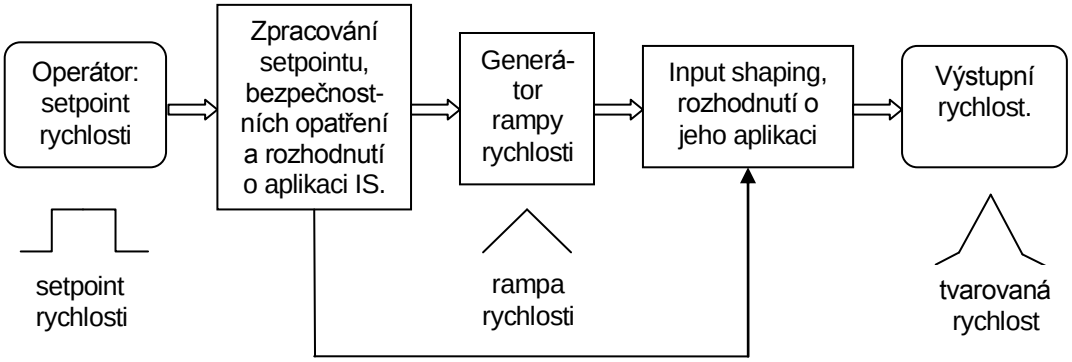
- Šířka zastavovací zóny ve směru pohybu kočky je 0,7m od kraje
- Šířka zastavovací zóny ve směru pohybu mostu je 1,5m od kraje
- Šířka zpomalovací zóny ve směru pohybu kočky je 4,8m od zastavovací zóny
- Šířka zpomalovací zóny ve směru pohybu mostu je 7,7m od zastavovací zóny

Velikosti těchto zón jsou bezpečné pro maximální délku lana, na kterém visí břemeno, rovné 12m. Tento rozměr by měl být maximální, kterým se běžně přejíždí s jeřábem. V případě, že by závěs byl delší, musí se input shaping počítat jen pro 12m, neboli tato hodnota musí být maximálním vstupním parametrem jako délka závěsu pro input shaper. Jinak by došlo k tomu, že systém bude ještě pomalejší a bezpečnostní zóny by přejel.

5.2 Implementace řídicího systému

Po dohodě s firmou SIMO-CZ bylo domluveno, že se input shaping bude aplikovat jenom na maximální rychlost, tzn. pokud operátor zadá pomocí joysticku maximální setpoint rychlosti. Důvodem je, aby mohl operátor přesně najet na požadované místo na nižších rychlostech a nevznikal zde problém, že operátorovi bude při tomto přesném polohování řídicí systém spíš na obtíž. Input shaping se tedy zapojí do řídicího systému, když bude potřebovat operátor přejet jeřábem plnou rychlostí a tedy většinou na dlouhé vzdálenosti.

Řídicí systém se skládá z několika bloků:



Obr. 5.3: Navrhované schéma řídicího systému

Generátor rychlosti na Obr. 5.3 vytváří, podle zadaných setpointů rychlosti, vstupní rychlost do input shaperu, který pak posílá tvarovanou rychlost na výstup. Generátor generuje tvar rychlosti, přesně podle požadavků v kapitole 3.2, jak je vidět na Obr. 3.2. Zároveň blok, který zpracovává setpoint, rozhoduje o tom, zda se má aplikovat input shaping a hlídá splnění všech bezpečnostních opatření. Rozhodnutí o aplikace input shapingu potom posílá přímo do shaperu.

Podle diagramu, provedu implementaci jednotlivých bloků v systému, všechny bloky jsou funkčně odzkoušené na modelu pomocí embbeded matlab function v Simulinku.

5.2.1 Blok zpracování setpointu rychlosti a bezpečnostních opatření

Tento blok má za úkol zpracovat příkazy (zde setpointy rychlosti) od operátora. Musí rozhodnout o tom, zda aplikovat input shaping a zda je jeřáb v ploše, ve které se smí pohybovat. Dále rozhoduje o tom, zda se má brzdit normální dobu, tedy 7s, nebo zda se má brzdit rychle - 4s, při vjezdu do zón zpomalení nebo zastavení.

Hlavička funkce, která reprezentuje tento blok:

```
function [ramp, setpoint, IS] = setpoint_rychlosti(joystick, joys-
tick_setpoint, mmax, ammax, abrzdmax, omezeni_rychlosti, stop, predch_setp)
```

Vstupní parametry funkce:

- joystick – poloha joysticku (např. hodnota 1 indikuje směr pohybu doprava a hodnota -1 doleva)
- joystick_setpoint – setpoint rychlosti od operátora
- mmax - maximální rychlost mostu
- ammax – zrychlení mostu
- abrzdmax – brzdné zrychlení ve zpomalovací a zastavovací zóně
- omezeni_rychlosti – indikuje polohu v zóně zpomalení (hodnoty 1,-1 jeřáb je v zóně, 0 jeřáb v zóně není)
- stop – indikuje polohu v zóně zastavení (hodnoty stejné jako u zóny zpomalení)
- predch_setp – setpoint rychlosti, který předcházel, než operátor zadal nový setpoint rychlosti (1 – pokud předcházel nejvyšší setpoint – 100%, 0 – pokud předcházel jiný setpoint)

Výstupní parametry funkce:

- ramp – zrychlení nebo zpomalení, které má generátor rychlosti použít pro vytvoření vstupní rychlosti do input shaperu. $\left[\frac{m}{s^2}\right]$
- setpoint – výsledný setpoint rychlosti
- IS – proměnná indikuje, zda se má použít input shaping (0 – bez input shapingu, 1 – s input shapingem)

Jak je ze vstupních parametrů poznat, tak následující kód je určen pro pohyb mostu. Pro pohyb kočky je kód podobný, jen se musí za vstup dosadit parametry pro kočku.

Vlastní kód:

```
function [ramp, setpoint, IS] = setpoint_rychlosti(joystick, joys-
tick_setpoint, mmax, ammax, abrzdmax, omezeni_rychlosti, stop, predch_setp)

setpoint=0; %výchozí hodnota setpointu rychlosti
IS=0; %výchozí hodnota - IS vypnut
ramp=ammax; %výstupní zrychlení je rovno zrychlení mostu

if joystick == 0 %joystick ve středové poloze
    setpoint=0; %setpoint rychlosti je nulový
    IS=1; %IS zapnut
end

if joystick == -1 %joystick doleva
    switch (joystick_setpoint) %nastavení velikosti setpointu rychlosti
                               %podle polohy joysticku
        case (1)
            setpoint=-0.1*mmax; %-10% maximální rychlosti
        case (2)
```

```
            setpoint=-0.25*mmax; %-25% maximální rychlosti
        case (3)
            setpoint=-0.5*mmax; %-50% maximální rychlosti
        case (4)
            setpoint=-mmax; %-100% maximální rychlosti
    end
    IS=1; % IS zapnut
end

if joystick == 1 %joystick doprava
    switch (joystick_setpoint) %nastaveni velikosti setpointu rychlosti
        case (1)
            setpoint=0.1*mmax; %10% maximální rychlosti
        case (2)
            setpoint=0.25*mmax; %20% maximální rychlosti
        case (3)
            setpoint=0.5*mmax; %50% maximální rychlosti
        case (4)
            setpoint=mmax; %100% maximální rychlosti
    end
    IS=1; %IS zapnut
end

if abs(setpoint) == mmax %test zda setpoint = maximální rychlost

    if omezeni_rychlosti*joystick == 1 %pokud jsou obě proměnné stejného
                                       %znaménka, pak jede jeřáb do zóny
                                       %omezení rychlosti

        if abs(stop) == 1 %test zda je jeřáb v zóně zastavení
            setpoint=0; %setpoint rychlosti je roven nule
            IS=1; %IS zapnut
            ramp=abrzdmax; %výstupní brzdné zpomalení je rovno brzdnému
                           %zpomalení v zóně zastavení
        else
            setpoint=sign(setpoint)*0.1*mmax; %setpoint rych. = max. ±10%
                                              %max. rychlosti
            IS=1; %IS zapnut
            ramp=abrzdmax;
        end

    end

else

if predch_setp == 1 %test zda předcházející setpoint byl maximální - 100%

    if (omezeni_rychlosti*joystick == 1 || abs(omezeni_rychlosti) == 1 &&
joystick == 0) % test zda najíždí jeřáb do zpomalovací zóny

        setpoint=sign(setpoint)*0.1*mmax;
        IS=1;
        ramp=abrzdmax;
    end

    if abs(stop) == 1
        setpoint=0;
        IS=1;
        ramp=abrzdmax;
```

```
    end

else %předcházející sepoint rychlosti nebyl maximální - 100%

    if omezeni_rychlosti*joystick == 1

        if abs(stop) == 1
            setpoint=0;
            IS=0; %IS vypnut
        else
            setpoint=sign(setpoint)*0.1*mmax;
            IS=0;
        end

    else
       IS=0;

    end

end

end
```

Z bezpečnostních důvodů jsem zakomponoval do tohoto bloku, že jakmile se jeřáb dostane do zóny zpomalení, tak se vždy testuje, zda už není v zóně zastavení, i když v některých případech by to nebylo potřeba, protože by tato situace neměla nastat.

5.2.2 Blok generování rychlosti

Tento blok má za úkol na základě generovaných setpointů rychlosti z předchozího bloku vytvořit profil rychlosti, který se buď bude zasílat přímo do měniče, nebo který se bude zpracovávat input shaperem. Funkce tohoto bloku je dobře znázorněna na Obr. 3.2.

Hlavička funkce, která reprezentuje tento blok:

```
function y = fcn(Ts, ramp, setpoint, yzpozd)
```

Vstupními parametry jsou:

- Ts – vzorkovací frekvence
- ramp – zrychlení zaslané z předchozího bloku
- setpoint – setpoint rychlosti zaslaný z předchozího bloku
- yzpozd – výstupní rychlost, zpožděná o jeden vzorek

Výstupním parametrem je:

- y – skalár, obsahuje vždy jeden vzorek rychlosti v aktuálním čase. Tyto vzorky generují křivku rychlosti jako na Obr. 3.2.

Vlastní kód:

```
function y = fcn(Ts, ramp, setpoint, yzpozd)

y=0; %výchozí hodnota je rovna nule
if round(setpoint*100)==round(100*yzpozd)%test zda se předchozí vzorek
                                         %rychlosti rovná současnému,
                                         %zaokrouhlením se vytváří
                                         %toleranční pásmo
    y=setpoint; %výstupní vzorek rychlosti je roven setpointu rychlosti
else
    if setpoint >yzpozd %test zda je setpoint větší než předešlý vzorek
    y=yzpozd+ramp*Ts; %výstupní vzorek vzniká přičtením diference rychlosti

    else
            if setpoint <yzpozd
                y=yzpozd-ramp*Ts;
            end
    end
end

end
```

5.2.3 Blok tvořící shaper a rozhodující o aplikaci input shapingu

Tento blok má za úkol vyhodnotit, zda se má aplikovat input shaping. Pokud vyhodnotí, že se aplikovat má, musí si pamatovat n předcházejících vzorků, aby mohl shaper spočítat tvar výstupního signálu. Velikost této paměti je závislá na délce závěsu a periodě vzorkování:

$$pamet = round\left(\frac{2\pi}{T_s\sqrt{\frac{g}{l_{max}}}\sqrt{1-\zeta^2}}\right) \tag{5.1}$$

Velikost paměti se vypočítá podle (5.1). Tato velikost znamená, kolik se musí do paměti uložit předcházejících vzorků signálu. Vhodné je spočítat velikost paměti pro maximální délku závěsu břemene l_{max}, tato velikost paměti pak bude stačit pro všechny ostatní případy.

Pro maximální délku závěsu $l_{max} = 12m$ a periodu vzorkování $T_s = 0{,}03s$ vychází velikost paměti 229 předchozích vzorků.

Hlavička funkce, která reprezentuje tento blok:

```
function vyst_rychlost = ei_shaper(rychlost, l, konstanty, pamet, IS)
```

Vstupními parametry funkce jsou:

- rychlost – vstupní rychlost
- l – délka závěsu břemene
- konstanty – pole:
    - konstanty(1) = perioda vzorkování

    - konstanty(2) = hmotnost břemena
    - konstanty(3) = tíhové zrychlení
    - konstanty(4) = koeficient tlumení
- pamet – pole předcházejících vzorků, velikost viz (5.1)
- IS – rozhodnutí o aplikaci input shapingu (IS = 1 – ano, IS = 0 – ne)

Výstupním parametrem je:

- vyst_rychlost - rychlost, která vstupuje do měniče (vzorky)

Vlastní kód:

```
function vyst_rychlost = ei_shaper(rychlost, l, konstanty, pamet, IS)

if l<12 %pokud je délka lana větší nebo rovna 12m, tak rovnou
        %vyst_rychlost = rychlost
if IS==1 %pokud je input shaping vypnut, pak vyst_rychlost = rychlost

Ts=konstanty(1); %přiřazení jednotlivých konstant z pole konstanty
ml=konstanty(2);
g=konstanty(3);
Cphij=konstanty(4);

omegan=sqrt(g/l); %výpočet vlastní frekvence
zeta=Cphij/(2*ml*omegan); %výpočet relativního tlumení
T=2*pi/(omegan*sqrt(1-zeta^2)); %výpočet periody tlumeného kmitání břemene

V=0.05; %tolerované zbytkové kývání (v %) na vlastní frekvenci kmitů

%velikost prvního impulsu shaperu
A1=0.2497+0.2496*V+0.8001*zeta+1.233*V*zeta+0.496*zeta^2+3.173*V*zeta^2;
%velikost třetího impulsu shaperu
A3=0.2515+0.2147*V-0.8325*zeta+1.415*V*zeta+0.8518*zeta^2-4.901*V*zeta^2;
%velikost druhého impulsu shaperu
A2=1-(A1+A3);
%poloha druhého impulsu v čase
T2=(0.5+0.4616*V*zeta+4.262*V*zeta^2+1.756*V*zeta^3+8.578*V^2*zeta-
108.6*V^2*zeta^2+337*V^2*zeta^3)*T;

%výpočet výstupní rychlosti po aplikaci IS (aktuální vzorek), diferenční
rovnice shaperu.
vyst_rychlost=A1*rychlost+A2*pamet(round(T2/Ts))+A3*pamet(round(T/Ts));

else
    vyst_rychlost=rychlost;
end

else
    vyst_rychlost=rychlost;
end
end
```

Výstupní rychlost tohoto bloku se může měnit skokově. K tomuto jevu dochází, když blok přepíná mezi signálem netvarovaným a tvarovaným a naopak. S tímto jevem by si měl

poradit frekvenční měnič a výsledný průběh vyhladit nastavením maximálního sklonu rampy rychlosti, kterou může akceptovat.

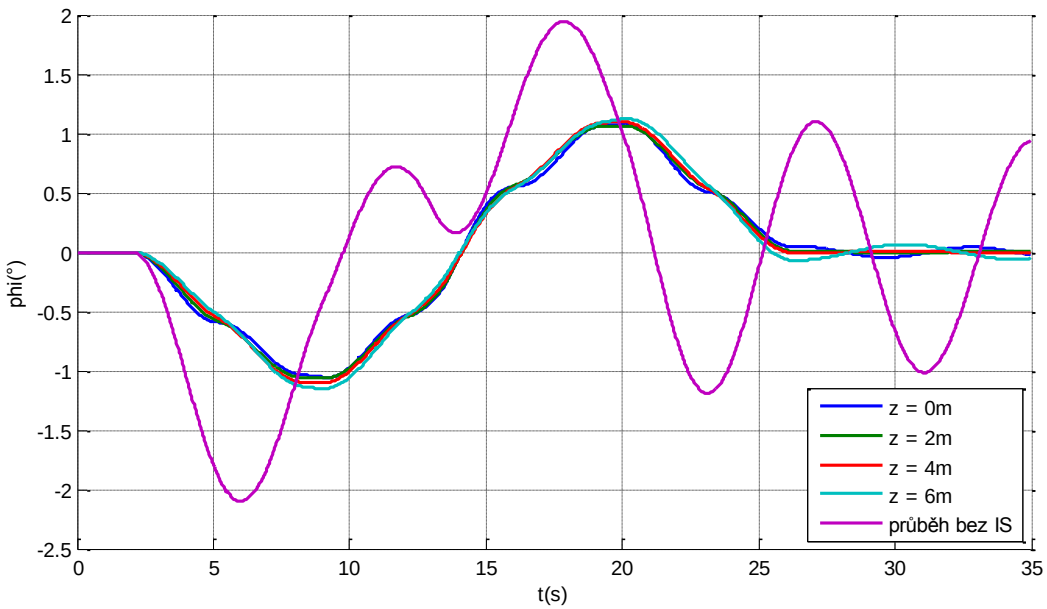
5.3 Testy řídicího systému

Pro otestování systému slouží přiložený model. Ovládá se pomocí joysticku. Setpointy rychlosti se nastavují pomocí tlačítek na joysticku. Model je ve virtuální realitě a tak je dobře vidět, zda jeřáb zastavuje kde má, tedy jestli je zajištěna bezpečnost provozu. V tomto modelu je řídicí systém implementován podle bloků v kapitole 5.2. Funkčnost bloků se dá tedy snadno ověřit pomocí tohoto modelu.

5.3.1 Robustnost systému vůči neznámé délce závěsu

Tento problém je zajímavý v tom, že ačkoli je měřena délka lana, na kterém visí kočka, není možné zjistit rozměry břemena, které bude zavěšeno na drapák. Proto provedu testy pro různě dlouhá břemena, která by mohla být zavěšena na drapák. Tím se zavádí chyba do systému a řídící program chybně určuje parametry pro polohu a velikost impulzů pro input shaper. Vzhledem k použití EI-shaperu, by si měl řídicí systém poradit i s celkem velkou chybou v určení délky závěsu.

Parametry testu jsou následující: délka lana $l = 10m$, vstupem do soustavy je rampa rychlosti podobná jako např. na Obr. 4.11, s tím rozdílem, že se začne brzdit o dvě vteřiny dříve. Jak vypadá vstupní rychlost, ale není při tomto testu důležité, proto vykreslím jen odezvu systému pro různě dlouhá břemena zavěšená na drapáku, viz Obr. 5.4.



Obr. 5.4: Robustnost řídicího systému vůči neznámé délce závěsu

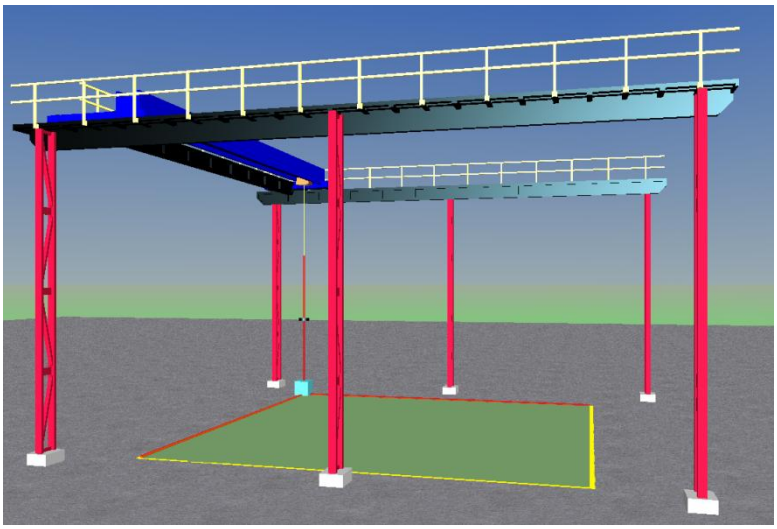
Jak je vidět na Obr. 5.4, tak délka závěsu nákladu z na drapáku nemá velký vliv na celkové rozkývání břemene. Ukazují se zde výhody velké robustnosti EI-shaperu.

Tato robustnost se bude také hodit při implementaci do reálného jeřábu vzhledem k tomu, že frekvenční měniče nastavují rychlosti motorů s určitou nepřesností a zpožděním. Celý systém bude zatížen tedy různými malými chybami a zde by měla robustnost EI-shaperu také pomoci.

5.3.2 Test užitečnosti a funkčnosti celého systému

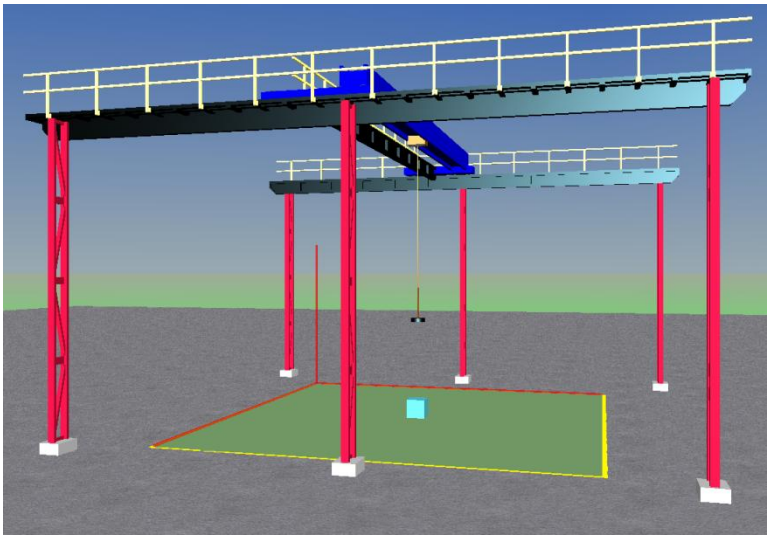
Testy provedu tak, že budu pomocí joysticku přesouvat břemeno z jednoho místa na jiné a budu měřit čas, který tento přesun zabere. Nejprve vypnu řídicí systém a zkusím přemístit břemeno pomocí joysticku, tak jak to funguje na skutečném jeřábu nyní. Poté zapnu navržený řídicí systém a opět zkusím přesunout břemeno úplně stejně, jako v předchozím případě. Břemeno bude umístěné na své pozici v tu chvíli, kdy bude výchylka kyvu maximálně $\pm 0{,}5°$.

Počáteční poloha břemene je vidět na Obr. 5.5:



Obr. 5.5: Počáteční poloha břemene

Koncová poloha břemene je vidět na Obr. 5.6:



Obr. 5.6: Koncová poloha břemene

Celkem provedu přejezd pětkrát a změřím jednotlivé časy přejezdů, výsledné časy jsou zapsány v tabulce Tab. 5.1. V testu budu počítat s tím, že systém měří délku lana závěsu, input shaper zná tedy vždy přesně vlastní frekvenci kmitů břemene.

měření	čas přejezdu s navrženým řídicím systémem (s)	čas přejezdu bez navrženého řídicího systému (s)
1	44,4	61,5
2	45,3	70,0
3	48,2	69,0
4	43,0	60,3
5	51,2	73,8

Tab. 5.1: Porovnání časů přejezdů s a bez řídicího systému

Jak je vidět dle Tab. 5.1, tak zvládne operátor přemístit břemeno rychleji v řízeném systému. Průměrný čas přesunu je $t_{řiz} = 46{,}42s$, oproti tomu v neřízené soustavě trvá přesun průměrně $t_{neřiz} = 66{,}92s$. Celková úspora času tedy je přibližně 31%. Tato hodnota by byla pravděpodobně trochu nižší, kdyby neřízenou soustavu ovládal zkušený operátor. Přesto je zde vidět veliký přínos řídicího systému.

6 Závěr

Cílem této práce bylo navrhnout vhodný řídicí systém pro řízení mostového jeřábu, který pomůže operátorovi při ručním řízení minimalizovat výkyv břemene.

Realizace započala tím, že bylo nutné vybrat vhodné přímovazební řízení, které zadanou úlohu splní. Zpětnovazební řízení bohužel nešlo použít, kvůli tomu, že reálný jeřáb nebývá vybaven senzorem snímajícím výchylku břemene. Nejvhodnějším typem řízení se po sérii testů ukázal input shaping. Jako shaper byl zvolen EI-shaper, protože disponuje velkou robustností na rozdíl od ostatních metod.

Všechny testy se realizovaly na modelu, který byl vytvořen v programu Matlab. V této práci bylo použito demo, které realizovalo portálový jeřáb ve virtuální realitě. Z tohoto dema byla použita pouze virtuální část. Část reprezentující dynamický model systému byla nahrazena systémem, který jsem vytvořil z dostupných materiálů a informací o skutečném jeřábu.

Vlastní řídicí systém je reprezentován jako program skládající se z několika bloků, které mezi sebou navzájem komunikují. Vlastní bloky jsou tvořeny jednoduchým kódem, který používá pouze aritmetické operace a podmínky, aby se dal řídicí systém jednoduše implementovat do PLC do reálného provozu.

Výhody anti-sway systému jsou především ve zkrácení doby transportu břemene z místa na místo. Je to způsobeno tím, že operátor může přemisťovat břemeno nejvyšší rychlostí, aniž by hrozilo rozkývání břemene. Další výhodou je, že po dojetí na požadované místo se břemeno nekývá a operátor může pomocí joysticku nejnižší rychlostí přesně umístit břemeno. Bez řídicího systému musí operátor jen svými schopnostmi zamezit kyvu břemene a to mu zabere více času.

Nevýhodou anti-sway systému je, že zpomaluje rychlost náběhu systému. Z toho plyne, že se operátor musí nejdříve naučit používat řídicí systém, aby zvládl včas zadat nulový setpoint rychlosti a zastavit na místě na kterém chtěl. Z testů vyplývá, že se požadované místo přejede, pokud na to není operátor připraven. Tato nevýhoda je však jen otázkou nácviku řízení. Větší nevýhodou je, že se kvůli pomalejšímu brzdění musejí zvětšit zóny zpomalení, popřípadě zastavení (u reálného jeřábu, pro který je tento systém navržen, se zóna zpomalení nemusí zvětšit, protože její velikost je dostatečná).

Jedinou veličinu, kterou potřebuje řídicí systém znát je délka lana, na kterém visí drapák. Toto měření bude nutné v případě reálného jeřábu doinstalovat.

V budoucnosti by se měl řídicí systém implementovat do PLC a testovat na reálném jeřábu firmy SIMO-CZ.

7 Citovaná literatura

[1] **Ahmad, M.A., a další.** *Investigations of Feed-forward Techniques for Anti-Sway Control of 3-D Gantry Crane System.* Pahang : Universiti Malaysia Pahang, 2009.

[2] **Huey, R.,John.** *The Intelligent Combination of Input Shaping and PID Feedback Control.* Georgia : Georgia Institute of Technology, 2006.

[3] **Singer, C. N. a Seering, P. W.** *Preshaping Command Inputs to Reduce System Vibration.* Massachusetts : MIT, 1988.

[4] **Singer, N., Singhose, W. a W., Seering.** *Comparison of Filtering Methods for Reducing Residual Vibration.* Massachusetts : European Journal of Control, 1999.

[5] **Singhose, W., Seering, W. a Singer, N.** *Residual Vibration Reduction Using Vector Diagrams to Generate Shaped Inputs.* . : Journal of Mechanical Design, 1994.

[6] **Piazzi, A. a Visioli, A.** *Optimal dynamic-inversion-based control of an overhead crane.* Parma : University of Parma, 2002.

[7] **Bartoš, Samuel.** *DP - Virtuálne laboratórium riedenia procesov.* Bratislava : Slovenská technická univerzita v Bratislave, 2006.

[8] **Franklin, Gene F., Powell, J. David a Emami-Naeini, Abbas.** *Feedback Control of Dynamic Systems.* New Jersey : Pearson Education, Inc., 2006.

[9] **HUMUSOFT s.r.o.** Portal Crane with Joystick Control. *Demo v Simulinku.*

[10] **Lang, Mathias C.** Least-Squares Design of IIR Filters with Prescribed Magnitude and Phase Responses and a Pole Radius Constraint. 2000.

8 Přílohy

Model ve virtuální realitě je na přiloženém CD.