

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

# MODELOVÁNÍ LINEÁRNÍHO POHONU PRO POTŘEBY NÁVRHU ŘÍZENÍ

MODELING OF LINEAR DRIVE FOR CONTROL DESIGN PURPOSES

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. JAKUB DRÁBEK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. PAVEL HOUŠKA, Ph.D.

BRNO 2010



## **ZADÁNÍ ZÁVĚREČNÉ PRÁCE**



## **ABSTRAKT**

Práce se zabývá problematikou návrhu řízení lineárního pohonu s využitím propojení mezi nástroji grafického programovacího prostředí NI LabVIEW a 3D CAD nástrojem SolidWorks. V práci jsou popsány možnosti virtuálního modelování mechatronických soustav a simulace v CAD systémech. Řízení je navrženo pomocí simulačních nástrojů s využitím propojení NI LabVIEW a SolidWorks a následně je navržené řízení vyzkoušeno na reálné soustavě.

## **ABSTRACT**

Thesis deals with the design of control of linear drive by taking advantage of connection between tools provided by NI LabView graphical programming environment and SolidWorks 3D CAD system. This work describes the various resources the CAD systems provide for virtual modeling of mechatronic systems and their simulation. Control is designed using simulation tools by utilizing link between NI LabView and SolidWorks, while the invented control mechanism is finally tested on real system.

### **Klíčová slova**

SolidWorks, SolidWorks Motion, NI LabVIEW, NI SoftMotion for SolidWorks, simulace

### **Keywords**

SolidWokrs, SolidWorks Motion, NI LabVIEW, NI SoftMotion for SolidWorks, simulation



## **PODĚKOVÁNÍ**

Děkuji svému vedoucímu Ing. Pavlu Houškovi, Ph.D za odborné vedení práce, z hodnotné rady, které mi poskytl na konzultacích a také za lidský přístup a pomoc, kterou mi poskytl při zpracování diplomové práce.

Dále bych chtěl poděkovat Ing. Janu Fridrichovskému za poskytnutí toolkitu.

V neposlední řadě děkuji své ženě Elišce za korekturu práce a za podporu.





## OBSAH

ZADÁNÍ ZÁVĚREČNÉ PRÁCE .....	3
ABSTRAKT .....	5
PODĚKOVÁNÍ .....	7
<b>1 ÚVOD .....</b>	<b>11</b>
<b>2 PRINCIPY MODELOVÁNÍ A SIMULACE MECHATRONICKÝCH SOUSTAV .....</b>	<b>13</b>
2.1 Návrh soustavy pomocí modelů .....	14
2.2 Postup návrhu soustavy založený na modelu .....	16
2.3 Stewartova platforma .....	18
2.3.1 Projekt Stewartova platforma VUT .....	18
<b>3 PŘÍSTUPY K VIRTUÁLNÍMU MODELOVÁNÍ.....</b>	<b>21</b>
3.1 Stavový prostor (State-Space) .....	21
3.1.1 Stavové rovnice.....	22
3.2 Tvorba modelu .....	22
3.2.1 Textové nástroje pro tvorbu modelu .....	22
3.2.2 Grafické nástroje pro tvorbu modelu .....	23
3.3 Simulace v systémech CAD .....	25
3.3.1 Přednosti virtuálních prostorových modelů .....	25
3.3.2 Analytické výpočty .....	25
3.3.3 Simulace v 3D CAD systémech.....	26
3.3.4 Výpočtové metody .....	26
3.4 Simulace pohybu v 3D CAD systému Solidworks .....	28
3.4.1 Realizace pohybové studie za pomoci SolidWorks Motion .....	28
3.4.2 Typy pohybových studií .....	29
3.4.3 Přehled prvků pohybové studie.....	30
3.5 Identifikace systémů.....	32
3.5.1 Rozdělení identifikačních metod .....	32
3.5.2 Postup pro identifikaci .....	33
3.5.3 Metoda odhadu parametrického modelu.....	34
3.5.4 Metoda odhadu částečně známého modelu .....	36
<b>4 REALIZACE MODELU V NI SOFTMOTION MODULE FOR SOLIDWORKS..</b>	<b>39</b>
4.1 Vlastnosti NI SoftMotion Module for SolidWorks.....	39
4.2 Postup návrhu SoftMotion module .....	40
4.3 Popis jednotlivých bloků NI SoftMotion module .....	41
4.4 Realizace virtuálního modelu Lineárního pohonu .....	48
4.5 Připojení realizované pohybové studie do NI LabVIEW .....	49
4.6 Zhodnocení výhody-nevýhody .....	51

---

<b>5</b>	<b>MODELOVÁNÍ A SIMULACE POMOCÍ NI LABVIEW A SOLIDWORK API...</b>	<b>53</b>
5.1	Připojení pomocí toolkitu .....	53
5.2	Identifikace lineárního pohonu v prostředí LabVIEW .....	54
5.2.1	Použití metody odhadu částečně známého modelu.....	55
5.2.2	Použití metody ARX.....	56
5.3	Návrh řízení lineárního pohonu.....	57
5.3.1	Metoda LQR .....	59
5.4	Realizace řídicího algoritmu.....	59
5.5	Simulace řízení na virtuálním modelu.....	63
5.6	Realizace řídicího algoritmu na reálném modelu.....	66
5.6.1	Reálný model .....	66
5.6.2	Řízení reálného modelu .....	67
5.7	Zhodnocení .....	71
<b>6</b>	<b>ZÁVĚR.....</b>	<b>73</b>
<b>7</b>	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>75</b>

# 1 Úvod

V dnešní době je integrace mechanických soustav, elektrických prvků a informačních technologií se softwarovým inženýrstvím do jednoho systému stále častějším fenoménem. Tento trend dal za vznik novému specifickému odvětví, které se zabývá návrhem mechanické a elektronické části systému a implementací řízení, mechatronice. Cílem oboru mechatronika je optimální návrh mechatronické soustavy z pohledu mechanické, elektrické a softwarové části. Při tomto návrhu hrají důležitou roli vztahy mezi jednotlivými obory. Z toho vyplývá, že při mechatronickém návrhu bude potřeba speciální softwarové vybavení, jak pro mechanickou část realizující virtuální soustavu tak i pro část návrhu řízení a modelování elektrických dějů. S vývojem informačních technologií a nástrojů pro realizaci virtuálních modelů soustavy lze dosahovat větší přesnosti při modelování, zároveň dochází k eliminaci počtu chyb při řízení reálných soustav založených na zjednodušeném matematickém popisu mechanické části. V současné době se do 3D CAD nástrojů implementují pohybové mechanické studie, které simulují dynamiku soustavy. To usnadňuje přechod z virtuálního modelu k reálné soustavě. Konstruktorům mechatronických soustav se naskýtá možnost simulace chování virtuálního modelu, při využití nástrojů modelování dynamických dějů se snižuje potřebný počet prototypů pro vývoj mechatronické soustavy, čímž se také snižují finanční náklady.

Spoluprací firem NI LabVIEW a Dassault Systemes SolidWorks Corp. vznikl nástroj pro propojení 3D CAD aplikace SolidWorks s možností dynamických simulací na virtuálním modelu s nástrojem pro návrh a realizaci řízení NI LabVIEW. Toto propojení pod názvem NI SoftMotion for SolidWorks poskytuje ovládání simulace virtuálního modelu v SolidWorks z prostředí LabVIEW. Tento nástroj bude použit pro řešení této diplomové práce.

Teoretická část diplomové práce se zabývá základními principy modelování a simulace mechatronických soustav. Popisuje přístupy k virtuálnímu modelování 3D CAD aplikací se zaměřením na 3D CAD aplikaci SolidWorks. Následně jsou vysvětleny základní metody pro použití identifikace systému. Praktická část diplomové práce si klade za cíl vytvořit v prostředí SolidWorks 3D CAD model lineárního pohonu a realizovat dynamickou pohybovou studii. Pomocí propojení NI LabVIEW a SolidWorks provést identifikaci lineárního pohonu za pomoci NI System Identification Toolkit integrovaného v LabVIEW a z výsledků identifikace navrhnout řízení virtuálního modelu. Dále ověřit navržené řízení realizované v NI LabVIEW na řízení virtuálního modelu v SolidWorks. V závěru práce je popsáno jak použít navržené řízení pro řízení reálného lineárního pohonu a je provedena verifikace výsledků.

Dalším cílem této práce je vytvořit komplexního průvodce pro použití řešené problematiky propojení NI LabVIEW a SolidWorks a umožnit tak její použití pro řešení dalších projektů. Průvodce je realizován jak pro komerční toolkit NI Softmotion for SolidWorks, tak i pro rozhraní přistupující ke studii v SolidWorks prostřednictvím API. Toto rozhraní bylo vytvořeno v rámci diplomové práce Ing. Janem Fridrichovským v roce 2009 a v této práci bylo dále rozšířeno o další potřebnou funkcionalitu. Rozhraní nebo také propojení je označováno pod jménem toolkit.



## 2 Principy modelování a simulace mechatronických soustav

Velký rozmach informačních technologií v posledních desítkách let umožnil proniknout modelování a simulaci systémů jak do inženýrských oborů, tak do přírodních a společenských věd. Jestli se jedná o konstrukci zavěšení kol u traktoru nebo metabolismus u bakterií, modely reálných systémů jsou vytvořeny pro pochopení vnitřních stavů systému a možnosti předpovídání jejich chování. Simulace se využívají jako náhrada skutečných experimentů systémů, s čímž plyne řada výhod:

- Ve srovnání s reálnými experimenty jsou virtuální experimenty, tzn. simulování reálných systémů, z finančního hlediska méně nákladné a také méně časově náročné, protože virtuální model prototypu je výrazně levnější variantou než vytvoření skutečného prototypu.
- Některé stavy systému nemohou být uskutečněny v reálném systému.
- Většinou jsou všechny aspekty virtuálních experimentů opakovatelné.
- Simulované modely lze většinou zcela ovládat, všechny vstupní proměnné a parametry systému můžeme předem určit. Toto není zcela možné u reálného systému.
- Simulované modely jsou také plně monitorovatelné, Všechny výstupní proměnné a vnitřní stavy můžeme sledovat, zatímco u reálných systémů by každá výstupní proměnná musela být sledována zvlášť, což vede k zvýšení nákladů. Každé měření také nepříjemně ovlivňuje chování systému.
- V některých případech není možné experimenty uskutečnit z morálních důvodů, jako například experimenty na lidech v oblasti zdravotnické techniky.

Modelování a simulace systémů nepřináší pouze výhody. Při tvorbě modelu musí být použita různá zjednodušení a nahrazení komplexních částí modelované soustavy, zanedbání nebo zjednodušení některých vlivů a vazeb z mnoha důvodů. Tudiž model jenom výjimečně přesně odpovídá modelované soustavě. Proto je nutné model verifikovat.

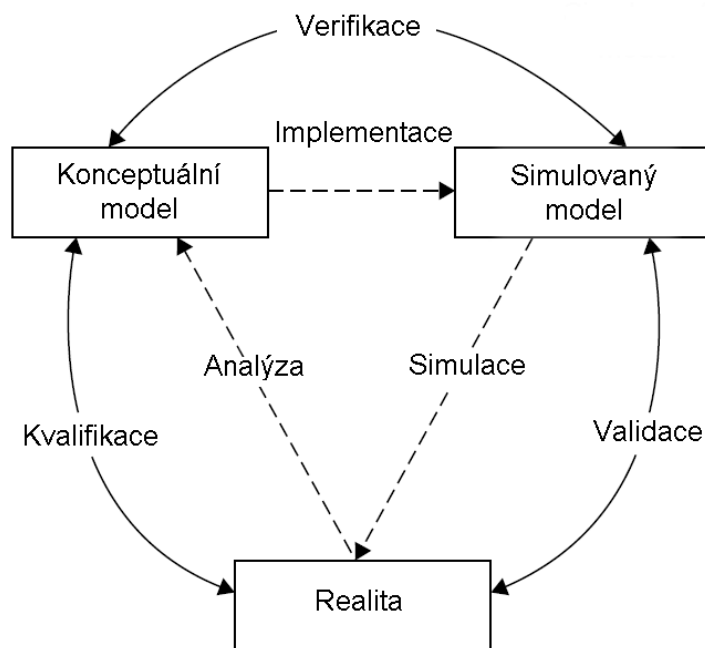
Přesnost hodnot, parametrů a rychlost simulace je závislá na výkonu počítače používaného pro simulaci.

Ve většině případů výhody zcela převažují a skutečné experimenty jsou nahrazovány virtuálními. Opakovatelnost virtuálního experimentu je zaručena počítačem a je zvláště výhodná také pro optimalizaci.

Dále budeme definovat řadu podmínek týkajících se modelování a simulace. Realita, jako jsou entity, situace nebo systémy, bude vyšetřována simulací. Modelování lze považovat za dvoufázový proces, který můžeme vidět na (obr. 1). První etapa, realita, je analyzována a modelována pomocí verbálních popisů, rovnic, vztahů nebo přírodních zákonů, které vytváří konceptuální model. Oblast použití je potom definována pro tento konceptuální model, který by měl poskytnout přijatelné reprezentace reality. Komunikace mezi konceptuálním modelem a realitou musí být definována pro vybrané oblasti použití.

V druhé etapě modelování je konceptuální model transformován do spustitelného souboru. Ten vytvoří soubor instrukcí, které popisují reakce systému na vnější podněty. Instrukce mohou být zpracovány ručně nebo pomocí počítače. Pozdější simulace umožňuje zpracování výrazně většího množství dat a zpracování složitějších problémů.

Vývoj modelů pro simulaci je složitý proces a značně náchylný k chybám. To způsobuje, že spolehlivost simulace je velmi závislá na kvalitě modelu. Metody a nástroje mají schopnost validace a verifikace. Verifikace modelu určuje, zda simulovaný model odpovídá konceptuálnímu modelu ve stanovených mezích přesnosti. Jinými slovy verifikace kontroluje korektnost modelu. Validace by měla určit, zda je simulovaný model vhodný pro plnění plánovaných úkolů v dané oblasti aplikace.[1]



Obr.1 Schéma realizace simulace.[1]

Mechatronika je vědní obor, který se zabývá projektováním moderních inteligentních produktů postupem „shora dolů“. Je postaven na strategii vycházející ze specifikace projektovaného produktu a zkoumání problémů s většími detaily. Pro návrh mechatronických systémů je nutnost nejen integrace poznatků z oborů jako jsou mechanika, elektrotechnika a informatika, ale i z ostatních oborů. Modelování hraje důležitou část mechatronice. V zásadě většina mechatrických návrhů je postavena modelování.

Význam slova mechatronika je odvozen ze spojení slov Mechanical systems a Electronic. Za důležitou část nového oboru byla považována také nauka o strategii plánování výroby a metodologii projektování mechatronických výrobků.

Díky rychlému rozvoji a dalšímu vylepšování akčních členů a senzorů, díky výkonné elektronice, integrovaným obvodům, mikroprocesorům, vysokému výkonu CAD modelování a výpočetní technice je disciplína mechatronika velmi významná.[2]

## 2.1 Návrh soustavy pomocí modelů

Abychom docílili vysoké efektivity vývoje je v dnešní době předběžné modelování mechatronických soustav nezbytností. Při formulaci modelů pro veškeré části soustavy je

potřeba využít všechny znalosti oborů, které jsou nutné pro návrh produktu. Vznikající modely popisují soustavu z několika aspektů. Typy modelů mohou být následující: model požadavků, představující požadavky na soustavu a model chování, charakterizující činnost soustavy. Modely, které popisují tvar produktu, jsou modely CAD. Nejdůležitější při návrhu modelování mechatronických soustav jsou modely popisující chování, protože formulují mezioborové funkční vazby. Celková činnost mechatronické soustavy je způsobena vzájemným působením všech technických oborů, proto je zapotřebí pracovat s určitými modely jednotlivých oborů. Na základě matematického popisu je integrace modelů při návrhu mechatronické soustavy jednoduchá a flexibilní a poněvadž je tento způsob velmi často používán, byly vytvořeny standardizované matematické popisy pro jednotlivé obory. Tento způsob návrhu mechatronických soustav založený na modelu přináší nezanedbatelnou úsporu času a nákladů. Aby bylo možné charakterizovat chování systému s určitou přesností, jsou modely vytvářeny na odlišných úrovních abstrakce. Základní modely jsou:[2]

a) Topologický model

Jde o základní modelování topologie systému, který bude simulován. Topologie systému popisuje uspořádání a propojení jednotlivých členů. Každý prvek modelu znázorňuje tři základní činnosti: kinematickou, dynamickou a mechatronickou. Kinematiku mechatronické soustavy stanovuje topologie mechanických částí.

b) Fyzikální model

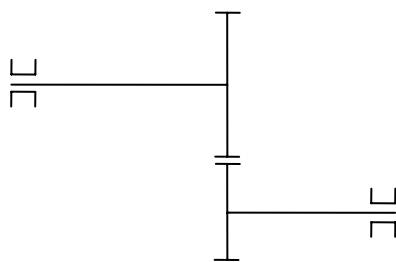
Model je vytvořen na základě topologického modelu a je definován vlastními proměnnými, jako je délka, hmotnost nebo pro elektrické systémy odpor a indukčnost. U mechanických systémů je například definován počet a propojení nepohyblivých částí, určení pohyblivých částí, tření, velikost ložisek a další. Fyzikální model charakterizuje systém v oborově-specifické formě.

c) Matematický model

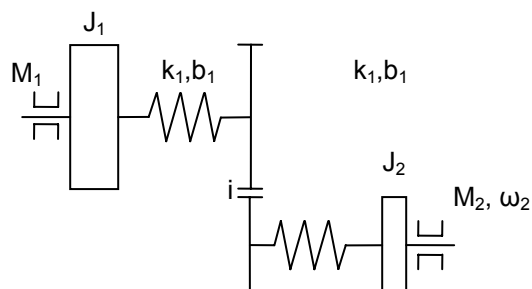
Jde o základní popis chování systému. Vychází z fyzikálního modelu, který se převede na abstraktní vyjádření ve formě matematického popisu, většinou jde o soustavu rovnic. Matematické modely mohou být odlišně vytvořeny, záleží především na pohledu uvažování nad podrobnostmi problému. Tím je například myšleno, zda do modelu zahrneme detaily popisu jako tření nebo popis pružných deformací a podobně.

d) Numerický model

Získáme ho z úpravy matematického modelu, aby bylo možné provést jeho algoritmizaci a simulaci na počítači. Numerický model je závislý na podrobnosti modelování předchozích modelů, metodě řešení a na matematickém modelu výhradně s ohledem na nelinearity systému. Poté jsou numerickému modelu zadány dané hodnoty. Tento proces je nazván parametrizací modelu. Následně je proveden výpočet modelu.[2]



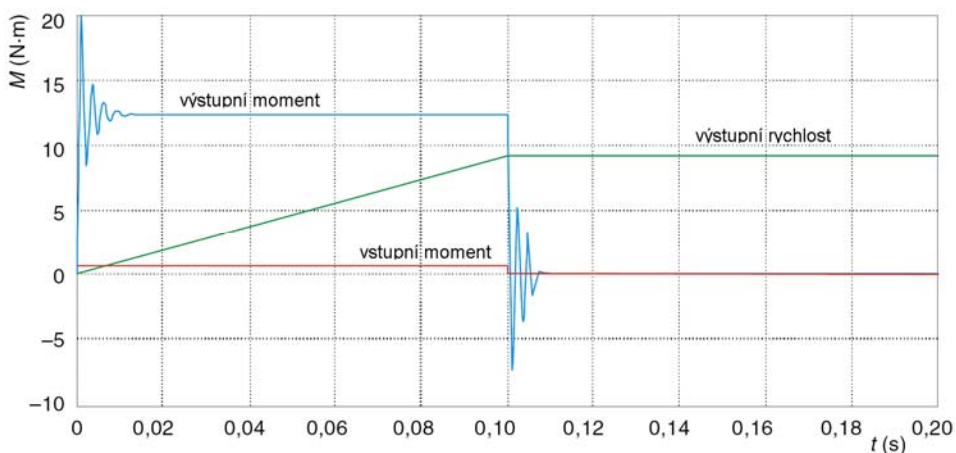
Obr. 2 Topologický model.



Obr. 3 Fyzikální model.

Matematický model je znázorněn v rovnici (2.1).

$$J_1 \frac{d\omega_1}{dt} = M_1 - M_{i1} - \frac{M_2}{i} \quad J_2 \frac{d\omega_2}{dt} = M_2 - M_{i2} \quad (2.1)$$



Obr. 4 Numerický model.[24]

## 2.2 Postup návrhu soustavy založený na modelu

Postup návrhu založený na modelu lze rozdělit do následujících pěti fází: Úloha, Modelování, Analýza modelu, Syntéza soustavy, Analýza soustavy.[1]



### 1. Úloha (Cíl)

Zpočátku je zapotřebí určit cíle a úkoly, které umožňují vybrat vhodný druh modelování. Druhy modelování mohou být následující:

- Zkoumání nové mechatronické soustavy (analýza návrhu řešení, požadavky pro orientovaný návrh).
- Návrh řídicího členu (lineární nebo nelineární analýza a syntéza pro obor řízení).
- Analýza a optimalizace již existující mechatronické soustavy.
- Nákladné, nebezpečné nebo nespolehlivé experimenty přímo na soustavě.
- Časově nákladné experimenty.
- Simulace pro snížení nákladů na prototyp.
- Simulace hardware pro ověření korektního chování.

Činnosti první fáze jsou následující:

- Provedení analýzy požadavků pro potřeby modelování.
- Určení cílů a úloh.

### 2. Modelování

Přesnost modelu je spojena s přesností výsledků analýzy. Výsledky analýzy modelu natolik odpovídají skutečnému modelu, nakolik model věrně popisuje systém. Činnosti v této fázi jsou popsány níže:

- Vytvoření modelu systému.
- Získání matematického modelu (analyticky nebo experimentálně).
- Uskutečnění identifikace parametrů.
- Testování a potvrzení modelu.

### 3. Analýza modelu

Za pomoci modelu jsou zjišťovány vlastnosti a chování systému. Např. metodou konečných prvků můžeme provést analýzu napětí. Poznatky z analýzy modelu jsou dále využívány pro fázi syntézy. Činnosti fáze analýzy modelu:

- Modifikace modelu pro simulaci.
- Analýza vlastností a chování systému.

### 4. Syntéza soustavy

Simulační a výpočtové výsledky analýzy modelu jsou aplikovány na danou soustavu. Optimalizují se principy činnosti a prvky řešení. Činnosti jsou následující:

- Určení přesných rozměrů, nastavení nebo optimalizace soustav.
- Nastavení parametrů soustavy.

### 5. Analýza soustavy

Optimalizovaná nebo zadaná soustava je opět analyzována a vyhodnocena. Z vyhodnocení může vyplynout návrat k předchozím fázím. Činnosti analýzy soustavy:

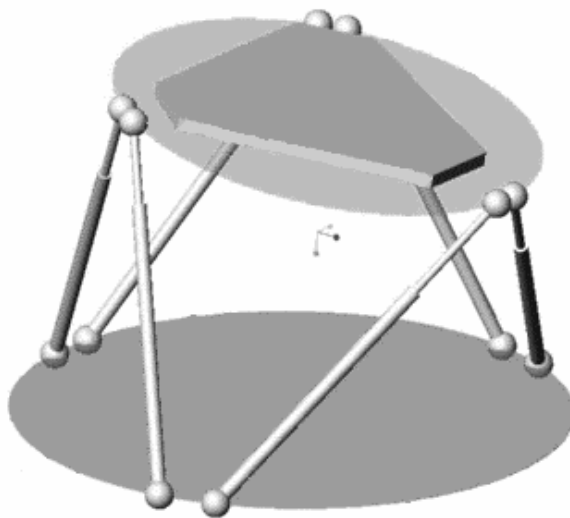
- Analýza a ohodnocení soustavy.
- Modifikace soustavy.[1]

## 2.3 Stewartova platforma

Stewartova platforma je popisována jako mechanismus s 6 stupni volnosti (6 Degrees-Of-Freedom). Původně byla Stewartova platforma navržena v roce 1965, jako letecký simulátor. Tato koncepce je v dnešní době stále využívána. Její návrh je přisuzován D. Stewartovi, který publikoval její návrh v The Institution of Mechanical Engineers (Stewart, 1965).[10] Doposud bylo navrženo velké množství aplikací, které čerpají z koncepce Stewartovy platformy, jako již zmíněný návrh leteckého simulátoru nebo obráběcí stroje či robotické manipulace. Využití najdeme také v oblasti námořnictví, automobilového průmyslu nebo lékařství. Platforma byla použita například k simulaci letu, modelu lunárního vozítka, stavění mostů, konstrukce mechanismu jeřábu atd.

Stewartova platforma má velký rozsah pohybu a může být s velkou přesností a snadností umístěna do požadované pozice. Většina variant Stewartových platform má 6 lineárně ovládaných akčních členů (nohou) s různými kombinacemi připojení k platformě. Je to paralelní mechanismus neboli uzavřený kinematický řetězec, který se skládá z horní desky, akčních členů a základní desky. Nohy jsou s horní a základní deskou spojeny 6 univerzálními klouby.[11]

Paralelní připojení má oproti sériovému mnoho výhod, jako jsou vyšší tuhost, vyšší únosnost a vyšší přesnost polohování. Existují ovšem i nevýhody, jako například omezení pracovního prostoru, menší manévrovatelnost a rozsah pohybu, velké zatížení v blízkosti singulárních bodů a další.[12]



Obr. 5 Schéma Stewartovy platformy.

### 2.3.1 Projekt Stewartova platforma VUT

V rámci spolupráce mezi Ústavem automatizace a informatiky, Ústavem mechaniky těles, mechatroniky a biomechaniky a Ústavem výrobních strojů, systémů a robotiky Fakulty strojního inženýrství VUT v Brně vznikají projekty zabývající se biomechanickými problémy.

Projekty se týkají zejména problémů klinické praxe. Rozvoj biomechaniky je v dnešní době podporován především výpočetní technikou a numerickými metodami pro řešení úloh, pro verifikaci výsledků simulačního modelování jsou však zapotřebí reálné experimenty.

Projekt, který vznikl spoluprací ústavů, je testovací zařízení, vycházející z konstrukce Stewartovy platformy. Tato konstrukce přibližuje základní pohyby určitých členů lidského těla. Lze zde konat obecný pohyb, získávat základní závislosti deformací, opotřebení určitých členů obecného a silového působení. Zařízení je určeno pro zjištění zatěžujících prvků (síla, silová dvojice, rozložené silové působení, posuv a úhel natočení) na daných částech lidského těla, na kyčelním kloubu (dřívku, jamce) a páteřní části (obratlích, ploténkách). Stewartova platforma se skládá z pohyblivé a pevné základní platformy. Platformy jsou mezi sebou spojeny šesti lineárními akčními členy (nohami), které jsou uchyceny sférickými vazbami. Nejdůležitějším a nejnáročnějším úkolem byl návrh lineárního akčního členu, jehož cílem je plnit základní požadavky aplikace, které jsou následující pomalý pohyb a velmi přesné polohování při velké zatěžující síle na platformu. Těchto požadavků se nedosáhne použitím lineárního elektromotoru. Proto byl vyvinut lineární pohon skládající se z elektrické pohonné jednotky zn. Maxon s motorem RE35, jednostupňovou planetovou převodovou skříní GP32C a snímačem otáček MR ENC typ L, dále čelního ozubeného soukolí a kuličkového pohybového šroubu s maticí. Kuličkový šroub s maticí typu K 12x4-3AP vyrobila firma Kuličkové šrouby Kuřim, a.s.[13] Zkonstruovaná Stewartova platforma s lineárními akčními členy je na (obr. 6).



*Obr.6 Stewartova platforma zkonstruovaná na Fakultě strojního inženýrství.*

Řízení je navrženo jako dvouvrstvé. První vrstva řízení je určena pro momenty na hřídeli u všech šesti pohonů, které zajišťují požadované natočení matic kuličkových šroubů a tím pádem polohu a orientaci horní platformy. Druhá vrstva je pouze řízení momentů na jednotlivých pohonech, kde vstupem je napětí. Pro druhou vrstvu řízení bylo z důvodu cenově přijatelnějšího získávání měřené veličiny navrženo polohové řízení s pozorovatelem. Druhá vrstva byla realizována koncepcí stavového řízení s kompenzací poruchy.[13]



### 3 Přístupy k virtuálnímu modelování

#### 3.1 Stavový prostor (State-Space)

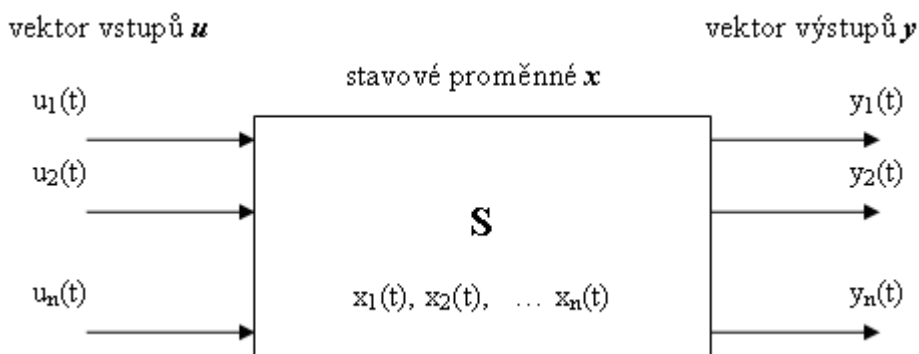
Stavový prostor je reprezentován jako matematický model fyzického systému se vstupy, výstupy a vnitřním stavem. Stavový prostor popisuje dynamické chování soustavy pomocí diferenciální rovnice prvního řádu a vnitřních proměnných, tzv. stavových proměnných. Vstupy, výstupy a stavy jsou vyjádřeny jako vektory nebo pomocí diferenciálních nebo algebraických rovnic, které tvoří spojení mezi stavovými a fyzikálními proměnnými. Z toho vyplývá, že jednotlivé stavové proměnné slouží pro reprezentaci fyzikálních veličin, jako napětí, proud, moment, rychlost apod.

Tento pohled nám umožňuje pozorovat vlastnosti chování daných dynamických charakteristik. Stavové proměnné jsou časově závislé proměnné definující chování systému v následujícím časovém kroku.

Níže jsou stavové veličiny znázorněny jako vektory. Vektor s označením  $u(t)$  je vstupní vektor,  $x(t)$  je označení pro stavový vektor a  $y(t)$  pro výstupní vektor.[15]

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \dots \\ u_n(t) \end{bmatrix} \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_n(t) \end{bmatrix} \quad y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_n(t) \end{bmatrix} \quad (3.1)$$

Soustavu lze zobrazit za pomoci blokového schématu, které je zobrazeno na (obr. 7). Jsou zde zobrazeny vektory  $u$  vstupující do systému, systém je popsán stavovými proměnnými  $x$  a ze systému vystupují vektory  $y$ .



Obr. 7 Schéma soustavy popsané stavovými proměnnými.

Počet vnitřních stavových proměnných je co nejmenší podmnožinou systémových proměnných, které reprezentují celý stav systému v daném čase. Minimální počet stavových proměnných  $n$  se většinou rovná řádu diferenciální rovnice. Jestliže je systém vyjádřen pomocí přenosové funkce, je minimální počet stavových proměnných řádu jmenovatele

přenosu. Převodem ze stavového prostoru na přenosovou funkci může dojít ke ztrátě některých vnitřních stavů systému.

### 3.1.1 Stavové rovnice

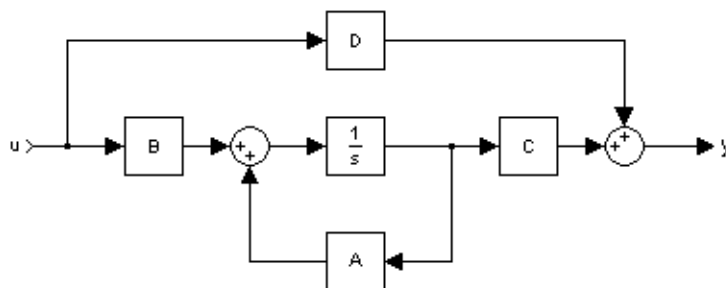
Nejčastější reprezentace stavového prostoru lineárního systému bývá vyjádřena v maticovém tvaru jako:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.2)$$

$$y(t) = Cx(t) + Du(t) \quad (3.3)$$

Rovnice (rce.3.2) se nazývá rovnice dynamiky, matice  $x$  je stavový vektor, matice  $A$  je označením pro matici systému, matice  $B$  je matice vstupu a  $u$  je vstupní vektor. Další rovnice (rce.3.3) se nazývá rovnice výstupu, kde vektor  $y$  je výstupní vektor, matice  $C$  je matice výstupu,  $D$  je matice přímé vazby výstupu na vstup. Rovnice dynamiky (rce.3.2) a rovnice výstupu (rce.3.3) jsou nazývány stavové rovnice spojitého lineárního systému. [16]

Stavové rovnice spojitého lineárního systému lze vyjádřit pomocí blokového schématu (obr. 8). Maticemi  $A$ ,  $B$ ,  $C$ ,  $D$  je plně určen vnitřní popis systému



Obr. 8 Blokové schéma stavových rovnic.

## 3.2 Tvorba modelu

### 3.2.1 Textové nástroje pro tvorbu modelu

Přehled základní používaných textových nástrojů pro tvorbu modelu je popsán níže.

#### Matlab m-file

Prostředí Matlab vyvinuté firmou The MathWorks umožňuje navrhovat modely v textové podobě. Nástroj pro práci v textovém editoru se označuje pod jménem m-file script (MatlabScript). Prostředí Matlab obsahuje dva základní typy m-file:

**Skripty (Scripts)** - M-file typu skript běží na pracovní ploše prostředí Matlab. Skript napsaný v příkazové řádce používá a manipuluje s proměnnými na pracovní ploše.

**Funkce (Function)** - Funkce se programují ve vlastním programovacím prostředí odděleném prostředí od pracovní plochy prostředí Matlab. Programovací prostředí s pracovní

plochou komunikuje a předává si potřebné proměnné. Pro složité algoritmy vyžadující mnoho proměnných tato koncepce může zapříčinit delší výpočty.[25]

### **NI LabVIEW MathScript**

Možností grafického programovací prostředí LabVIEW je také programování pomocí textového editoru, který se nazývá LabVIEW MathScript RT Module. Syntaxe kódu je stejná jako pro programování v m-file scriptu v prostředí Matlab. Výhodou tohoto modulu je možnost načtení m-file scriptu z Matlabu. Práce s LabVIEW a MathScriptem umožňuje efektivně vytvářet textové a grafické programy.[22]

### **GNU OCTAVE**

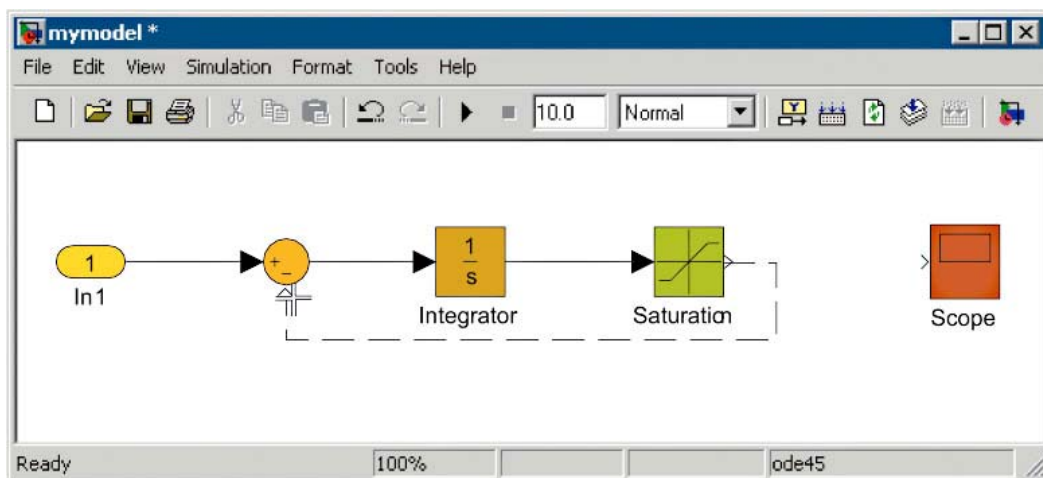
OCTAVE je vysokoúrovňový programovací jazyk určený pro numerické výpočty. Systém pro řešení lineárních rovnic, hledání kořenů nelineárních rovnic, integrování rovnic a další. Jazyk je kompatibilní s aplikací Matlab, kde se využívá. GNU Octane je volně rozšiřitelný a modifikovatelný podle podmínek GNU General Public Licence (GPL). Pomocí skriptů ve vlastním jazyce nebo pomocí dynamických modulů v C, C++, FORTRANU lze program jednoduše rozšířit do jiných aplikací.[26]

## **3.2.2 Grafické nástroje pro tvorbu modelu**

### **Simulink**

Simulink je software určený pro modelování, simulování a analyzování dynamických systémů. Simulink je integrován v prostředí MATLAB. Poskytuje grafické prostředí a knihovny, které umožňují navrhovat, simulovat, implementovat systémy, včetně komunikace, návrhu řízení a zpracování signálu. Podporuje lineární a nelineární systémy. Simulink obsahuje grafické uživatelské rozhraní pro vytváření modelů pomocí blokových schémat, které jsou umístěny v knihovně bloků. Interaktivní grafické prostředí zjednodušuje proces modelování, eliminuje potřebu formulovat v jiném programu či prostředí diferenciální a diferenční rovnice pro popis systému. Pomocí bloků umístěných v knihovně se vytvářejí modely systémů. Knihovna obsahuje bloky pro spojité a diskrétní systémy (integrátor, zpoždění), algoritmické bloky (sčítání, násobení, vyhledávací tabulka) a strukturální bloky (Mux, přepínač).

Modely se vytvářejí přetahováním bloků z knihovny a propojením s ostatními bloky, které vytvářejí matematické vztahy mezi bloky. Na (obr. 9) je znázorněna ukázka vytvoření modelu v grafickém prostředí Simulink.[21]



Obr. 9 Vytváření modelu v Simulink [21].

Simulink lze rozšířit o dodatečné nástroje, které obsahují speciální funkce například pro letectví, komunikace, rádiové frekvence, zpracování signálu a další aplikace. Je možnost modelování fyzických systémů s nástroji, jako jsou Simscape, SimDriveline, SimHydraulics, SimMechanics, SimPowerSystems, které zahrnují mechanické, elektrické a hydraulické komponenty.

### SimMechanics

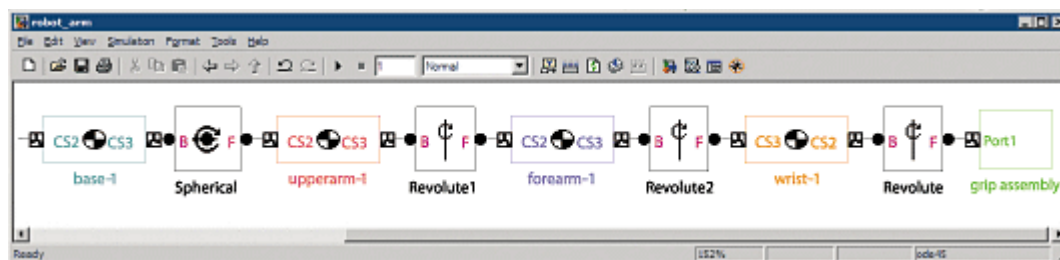
SimMechanics je nástroj sloužící pro modelování a simulování mechanických systémů. Patří do skupiny produktů Simulink Physical Modeling Family (simulace a modelování soustav), který je nástavbou programu MATLAB, vyvinutý společností The MathWorks. The MathWorks poskytuje software pro technické výpočty a modelování soustav.

SimMechanics je nástroj pro modelování 3D mechanických systémů, založený na Simscape v prostředí Simulink. Simscape je nástroj pro modelování systémů zahrnující mechanické, elektrické, hydraulické a další fyzické prostředky. Místo matematických rovnic popisujících soustavu, můžeme použít tento simulační nástroj, který umožňuje sestavovat model pomocí bloků odpovídajících reálným částem mechanismů, jako například tělesa, vazby mezi nimi, kloubová spojení, tlumiče, pružiny, senzory, akční členy, definici fyzikálního prostředí a dalších, tím jsou popsány fyzické komponenty a vztahy mezi nimi. Na základě definovaných parametrů a spojení bloků, je automaticky vytvořen matematický model pro Simulink. Například pomocí bloku Sensor můžeme fyzikální signály (polohu, rychlost atd.) převést na signály určené pro Simulink a dále s nimi pracovat. Dále je možnost je zpětně převést pomocí bloku Actuator na fyzikální signály (síla, moment atd.) a těmito signály potom působit na mechanismus.[20]

Následně je automaticky vygenerována 3D animace, která vizualizuje dynamiku systému. 3D modely můžeme nahrát z některých CAD systémů a tím převést vlastnosti modelu, jako jsou hmotnost, pohybové omezení, geometrie atd. do prostředí SimMechanics.[17]

Ukázka návrhu soustavy pomocí bloků SimMechanics je na (obr. 10). [17]





Obr. 10 Návrh soustavy bloky SimMechanics. [17]

### 3.3 Simulace v systémech CAD

Základním produktem moderních CAD systému jsou prostorové virtuální modely dílů a sestav. Tyto modely jsou detailně zkonstruovány, proto z nich lze vytvořit výkresovou dokumentaci jako souhrnné podklady pro výrobu.

Virtuální modely poskytují mnohem více možností a výhod. Při vývoji výrobku zásadně ovlivňují cenu prototypu a také dobu zavedení výrobku do sériové výroby. Je to způsobeno tím, že konstruktéři mají možnost provádět simulace kinematiky sestavy, vykonávat pevnostní nebo tepelné analýzy a jiné. Proto není potřeba vyrábět drahé skutečné prototypované díly a zkoušet jejich reálné vlastnosti.

#### 3.3.1 Přednosti virtuálních prostorových modelů

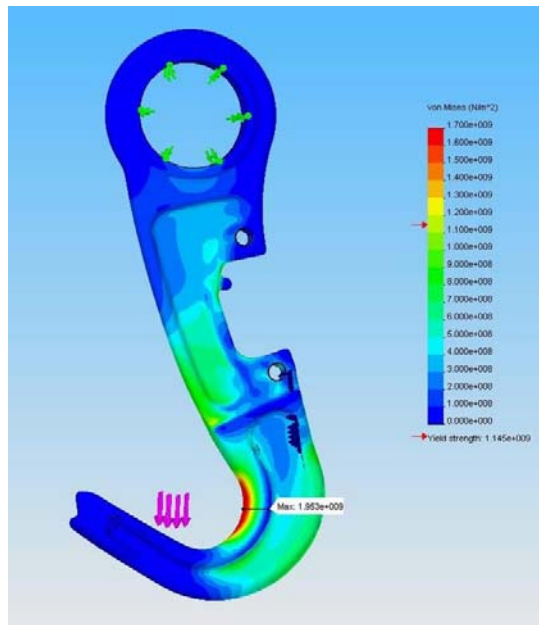
Moderní CAD systémy poskytují schopnosti tvorby virtuálních 3D modelů pro jakékoli odvětví průmyslu. Přednosti virtuálních prostorových modelů jsou následující:

- Jednoznačné vizuální pochopení a představení složitých dílů a sestav (nejčastěji využíváno u prezentací při obchodních jednání).
- Snadné vytváření výkresové dokumentace.
- Využití při rychlém prototypování (Rapid Prototyping) jako je tisk na 3D tiskárnách.
- Možnost analýzy dílů a sestav, což zkracuje dobu potřebnou k testování dílů při vývoji a také snížení finančních nákladů.[3]

#### 3.3.2 Analytické výpočty

Na začátku v systémech CAD byly výpočtové modely založeny na analytických metodách. Tyto metody byly poměrně rychlé, velkou nevýhodou však bylo značné zjednodušení virtuálních modelů, proto nebylo možno přesně vystihnout reálné chování dílů nebo mechanismů. Oproti tomu výhodou byla jednoduchost a rychlost výpočtu. V současné době se tyto metody výpočtu využívají pro všeobecnou představu u namáhání dílů, což pomáhá pro dimenzování dílů. Jako příklady do této skupiny výpočtového systému lze zařadit TDS-Technik nebo také Mechsoft Profi. V těchto programech lze provádět především pevnostní kontrolu normalizovaných součástí (šrouby, čepy, kolíky, nýty, klíny, pera, pružiny atd.) nebo také výpočty šroubových spojů, svarů, výpočty nosníků staticky určitých a

neurčitých. Výhodou výpočtu výsledků jsou rychlé informace o tom, zda díl vyhovuje daným požadavkům. Systém mu dále může navrhnout optimalizované řešení příkladu anebo poskytné další normalizovanou součást. [3]



Obr.11 Grafické znázornění tažného působení na hák. [3]

### 3.3.3 Simulace v 3D CAD systémech

Velké CAD systémy mají většinou integrované nebo také samostatně dostupné výpočtové systémy, jako jsou například COSMOS, ANSYS a to v podobě doplňkového modulu. Další možností implementace nástrojů pro analýzy jsou výpočtové moduly, které vytvářejí přímo firmy zabývající se CAD systémy, například Pro/Mechanica v CAD systému Pro/Engineer.

K řešení úloh v simulačních programech se využívají různé numerické metody. Ty mají společnou vlastnost, která spočívá v diskretizaci proměnných. Výhodou je možnost získání řešení úlohy v konečném počtu diskretních uzlů vygenerované sítě získané například z konečných prvků. Základní numerické metody jsou zejména Metoda konečných prvků a Metoda konečných diferencí. [3]

### 3.3.4 Výpočtové metody

Ve většině CAD systémů existují nástroje pro výpočetní analýzu. Využívají se pro řešení matematických úloh, postavených na Metodě konečných prvků (MKP, angl. FME) nebo na Metodě konečných diferencí (MKD, angl. FDP).

Výpočtové analýzy můžeme rozčlenit do následujících skupin:

a) **Statické analýzy**

Výsledkem výpočtu u lineárně-statické analýzy mohou být poměrné deformace, posuny, napětí, síly apod. Zatížení soustavy může být silou nebo momentem. U této analýzy není možné zadávat zatěžující proměnné v čase.

b) **Dynamické analýzy**

Analýza kinematiky pohybu (pohybová studie, Motion study), fyzikální analýzy, proudění.

Pomocí simulací je možno vypočítat také proudění kapalin, plynů nebo obtékání objektů, proudění v kanálech, potrubích vzduchotechniky atd. Většina simulačních modulů obsahuje pevnostní simulace pro pevnostní výpočty. Výsledky simulací jsou pro názornost reprezentovány v grafické formě, která je pro uživatele vhodná z hlediska rychlého posouzení vyhodnocení.

V následujícím textu jsou popsány možnosti simulačních nástrojů v CAD systému SolidWorks 2009 (SolidWorks Simulation):

### **Úlohy lineární statiky**

Konstantní zatížení nebo zatížení s pomalou změnou. Zatížení způsobuje deformaci v oblasti pružnosti materiálu. Hlavními výsledky řešení jsou napětí a deformace.

### **Teplotní analýza**

Pozoruje působení tepelných toků a teplotních polí na mechanické vlastnosti zatížených dílů. Výsledky výpočtu jsou zejména teplotní napjatost a dilatace. Jinými úlohami jsou analýza přenosu tepla vedením, prouděním atd.

### **Frekvenční analýza**

Součásti namáhané vibracemi nebo odstředivým zatížením.

### **Únavové analýzy**

Součásti nebo sestavy namáhané cyklickým zatížením, kde je možnost poškození během provozu. Využívá se úloh lineární statiky. Jeden zatěžovací cyklus je určen pro výpočet celkové životnosti. Dále se používá při analýze pádové zkoušky.

### **Optimalizace sestavení**

Výsledkem jsou materiálové úspory, zdokonalení vlastností dílů. K problému se přistupuje iterací nalezení vyhovujícího tvaru součástí.

### **Úlohy nelineární analýzy**

Nelineární stavy, ve kterých dochází k namáhání za mez kluzu materiálu nebo se vyskytují velké deformace v součástech. Systém dokáže řešit úlohy obsahující plasticitu, viskoelasticitu, elastické a teplotně závislé vlastnosti materiálu atd.

### **Simulace proudění**

Výpočty proudění plynů a kapalin. Možnost výpočtu stlačitelných a nestlačitelných médií uvnitř nebo vně potrubí. Dále řešení viskózního, laminární a turbulentního proudění nebo

také řešení proudění v rotačních součástích. Výsledky jsou zpracovávány grafickou formou.

### **Pohybová studie a simulace**

Z nástrojů simulace pohybu je možnost výběru ze dvou odlišných variant. První je určena pro snadnou a rychlou kontrolu kinematiky. Jedná se o základní testování pohybu dílů sestav v reálném čase, kde v průběhu simulace reálný čas není brán v úvahu. Tato simulace zanedbává složité prvky a bere v potaz pouze jednoduché, jako jsou například lineární pružiny, lineární nebo rotační motory. Také je možnost sestavu zatížit gravitací. Výsledkem řešení je pouze vizuální animace pohybu sestavy. Druhý způsob simulace pohybu je mnohem sofistikovanější. Nazývá se Solidworks Motion. Tato analýza pohybu je založena na fyzikální simulaci v reálném čase. Pohyb sestavy je řízen konstantními nebo proměnnými hodnotami polohy, rychlosti a zrychlení určitého uzlu nebo celé součásti. Pomocí vazeb v sestavě je možné určit vzájemné ovlivňování dílů v sestavě. Může se jednat o přenos pomocí vnějších lineárních a nelineárních sil, kloubů, pružin atd.

### **FEA převodníky**

Úlohou je schopnost sdílení datových souborů mezi jednotlivými CAE a CAD systémy. (ANSYS Translator) [3]

## **3.4 Simulace pohybu v 3D CAD systému Solidworks**

SolidWorks je 3D CAD program pro modelování a navrhování součástí a soustav, zahrnující celou řadu funkcí pro zlepšení a usnadnění modelování CAD modelů pro nejrůznější obory. Byl vyvinut francouzskou společností SolidWorks Corporation, nyní jako dceřiná společnost Dassault Systèmes. Jako většina 3D CAD systému i tento obsahuje tři základní typ dokumentu. První typ je pro vytváření základního dílu mechanismu, který se využívá pro další typ dokumentu. Tím je složení sestavy za pomoci jednotlivých dílů. Posledním typem dokumentu je vytvoření výkresu, který je důležitý pro výkresovou dokumentaci dílů a sestav pro jejich výrobu. Pomocí přídatných funkcí, tzv. „doplňkových modulů“, dokáže SolidWorks provádět také různé simulace a analýzy soustav (analýza a simulace proudění, napět'ová, teplotní analýza, pohybová simulace prostřednictvím modulu pod názvem SolidWorks Motion, dříve COSMOSMotion ).

### **3.4.1 Realizace pohybové studie za pomoci SolidWorks Motion**

SolidWorks Motion je nástroj pro vytváření kinematických a dynamických analýz modelu. Za pomoci výsledku analýzy zjistíme, zda projekt vyhovuje požadavkům, které jsme stanovili. Tento nástroj je založen na výpočtovém simulačním jádře MSC ADAMS, které je plně integrováno do prostředí SolidWorks.

Simulace pohybu poskytuje kompletní a kvantitativní informace z kinematiky, zahrnující informace o poloze, rychlosti a zrychlení soustavy a také z dynamiky, jako například společné reakce, setrvačné síly a energie od každého jednotlivého prvku soustavy. Výsledek simulace pohybu není nadměrně časově náročný, protože veškeré potřebné

parametry jsou již nadefinovány v modelu sestavy CAD, je pouze nutné převést tyto parametry do programu simulace pohybu. Program využívá vlastnosti materiálu z CAD soustavy a definuje setrvačnost mechanismu.[4]

Návrh mechanismu pro možnosti simulování prostřednictvím pohybové studie je následující. Nejdříve jsou vymodelovány základní díly mechanismu v prostředí SolidWorks. Každému dílu je definován materiál. Korektní zadání materiálu je velmi důležité pro přesnou realizaci simulace výpočtu. Dále se ze základních dílů vytvoří sestava s odpovídajícími vazbami. Vytvořený model sestavy s určenými vazbami dílů je automaticky převeden do dynamické pohybové studie. Dále je v pohybové studii možnost určení, které vazby budou akceptovány, a které budou potlačeny (tzn. díly jsou pevně spojeny). Dále je zapotřebí určit působení jednotlivých těles na ostatní. Mezi součásti je možno vložit pružinu, tlumič se zadanými parametry nebo vytvořit dotykovou vazbu mezi vybranými součástmi. Můžeme určit tzv. motor, který pohybuje součástí jako kdyby na ni působil motor. Prostřednictvím vazeb, které jsou určené, působí na další součásti sestavy moment nebo síla. Po výpočtu simulace studie lze znázornit reakční síly ve vazbách a také hodnoty, například posunutí, rychlosti, zrychlení, tlaku, momentu, energie a síly daného tělesa zobrazit v grafu.

V programu SolidWorks existují široký výběr druhů zadání vazeb. Typy vazeb jsou rozděleny do několika skupin. Ve skupině Standardní vazby lze vybrat z možností vazba sjednocená, rovnoběžná, kolmá, tečná, soustředná, zámek, vzdálenost a úhel. V další skupině pod názvem Upřesňující vazby jsou na výběr vazby symetrická, vystředění, vazba trajektorie, lineární vazba. Následující skupina Strojní vazby je pro pohybovou studii velmi významná. V této skupině jsou možnosti vačka, čep, převod, ozubená tyč, šroub a univerzální spoj. Při použití studie Motion Analysis k výpočtu pohybu, která bude podrobněji probrána níže, nástroj vypočítá počet stupňů volnosti v mechanismu a dokáže odstranit nadbytečné vazby. Automatické odstranění nadbytečných vazeb není vždy korektní, proto je cílem návrháře vytvořit model bez nadbytečných vazeb. Je také možnost nahradit nadbytečné vazby pouzdry. Potom je zatížení v systému rozloženo na základě toho, které vazby jsou nahrazeny. Tato varianta prodlouží dobu výpočtu pohybové studie.

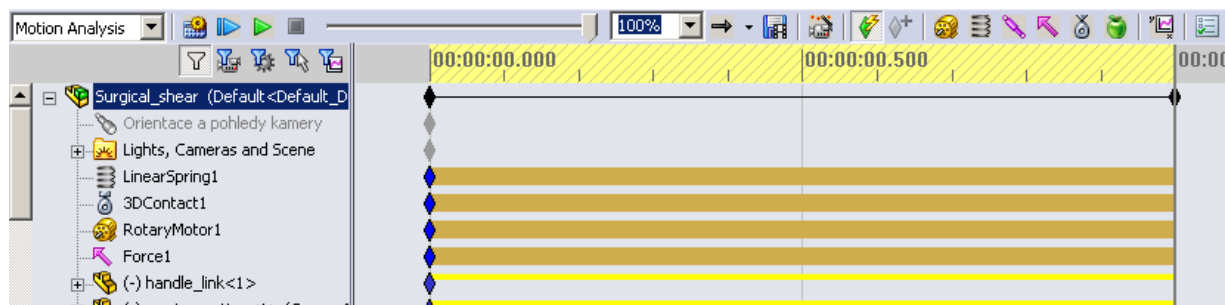
### 3.4.2 Typy pohybových studií

Pohybová studie graficky simuluje pohyb modelů sestavy. Nemění model sestavy, ani jeho vlastnosti, ale pouze animuje simulovaný pohyb modelu. K dispozici jsou vazby, které mohou omezit pohyb dílů v sestavě. Uživatelské rozhraní pohybové studie (obr. 12) pod názvem MotioManager je založeno na časové ose. SolidWorks ve verzi 2009 používá v nástavbě MotionManager tři rozdílné typy pohybových studií:

- a) Animace (angl. Animation, dříve Pohyb sestav)  
Pouze animace pohybu sestav. Animace používá interpolaci k určení pohybu součástí sestavy mezi klíčovými body. Vhodné pro animaci pohybu kde nejsou vyžadovány fyzikální vlastnosti nebo gravitace.

- b) Základní pohyb (angl. Basic Motion, dříve Fyzická simulace)  
Počítá se členy, jako jsou motory, pružiny, tlumiče, tlak, gravitace, atd. Zahnuje také výpočet kolizí se soustavou, fyzikální vlastnosti. Výpočet simulace je poměrně rychlý. Implementovaná technologie AGEIA PhysX.
- c) Motion Analysis (dříve COSMOSMotion)  
Používá se pro přesnou simulaci a analýzu prvků pohybu na sestavě. Do výpočtu zahrnuje působení sil, pružin, tlumičů, tření atd. Využívají se výpočetně náročné kinematické řešiče, které zahrnují vlastnosti materiálu, fyzikální vlastnosti a setrvačnost soustavy. Tento nástroj je výpočetně nejnáročnější a nejpřesněji vystihuje chování simulované soustavy.

Typy pohybových studií jako Animace a Základní pohyb jsou přímo implementovány v jádře SolidWorks. Nástroj Motion Analysis je dostupný v rámci doplňkového modulu SolidWorks Motion. Motion Analysis se využívá k propojení s prostředím NI Labview,[5]



Obr.12 Pohybová studie SolidWorks 2009.

### 3.4.3 Přehled prvků pohybové studie

V pohybové studii uskutečňujeme pohyb nebo působení různých vlivů na díl nebo sestavu jako jsou síla, pružina gravitace atd. pomocí tzv. Motion Study prvků. Dostupnost prvků v pohybové studii je rozdílná od daného typu pohybové studie, které byly vysvětleny výše. Prvky v pohybové studii je možno rozdělit do dvou základních skupin:

#### Skupina Motory

Pomocí prvků motory aplikujeme pohyb na součásti bez ohledu na setrvačnost a fyzikální vlastnosti. Pohyb, který je poháněn motorem nahrazuje veškerý pohyb způsobený jakýmkoliv jiným prvkem pohybové studie. Veškeré prvky, které odporují pohybu motoru, zvyšují spotřebu energie motoru, ale nesnižují jeho rychlost činnosti. Jestliže některý prvek zapříčiní změnu smyslu otáčení motoru, bude motor otáčet součást v jiném směru. Toto může nastat například, když dojde ke kolizi mezi součástí poháněnou motorem a odlišnou součástí. Pohyb motoru lze vybrat z definovaných typů motorů nebo použitím matematických výrazů. SolidWorks má na výběr ze dvou definovaných typů motorů:

#### Rotační motor

#### Lineární motor (Ovladač)

Při další specifikaci výběru motoru můžeme zvolit druh pohybu a jeho odpovídající hodnoty. Je zde na výběr následující:

- Konstantní otáčky – konstantní rychlost motoru.
- Vzdálenost – motor je v chodu do zadané vzdálenosti a za určitý čas (toto nastavení je nutné pro komunikace s prostředím LabVIEW).
- Kmitání – zadávají se hodnoty amplitudy a frekvence.
- Interpolovaný – řešič interpoluje splají mezi body dat.
- Výraz – (jen pro Motion Analysis).

### **Skupina Prvky síly**

Prvky síly nezabraňují pohybu, tzn. nepřidávají nebo neodstraňují stupně volnosti modelu. Síly mohou způsobit odpor pohybu nebo pomáhat v pohybu. Pohybová studie umožňuje výběr z následujících typů prvků síly:

#### **Gravitace**

Simulace gravitační síly (pouze pro Základní pohyb a Motion Analysis).

#### **Pružina**

Simuluje pohyb součástmi v celé sestavě různými druhy pružin. Druhy pružin:

- Lineární pružina (pouze pro Základní pohyb a Motion Analysis).
- Torzní pružina (pouze pro Motion Analysis).

#### **Tlumič**

Simuluje prvek tlumič v sestavě (pouze pro Motion Analysis). Druhy tlumičů:

- Lineární tlumič.
- Torzní tlumič.

#### **Síla**

Definuje zatížení a posunutí dílů v potřebném směru. Simulace pohybu u pohyblivých dílů v sestavě (pouze pro Motion Analysis). Typy sil:

- Lineární síla.
- Torzní síla.

#### **3D kontakt**

Definování kontaktu mezi součástmi, u nichž dochází ke styku. Je možno definovat materiál, tření mezi součástmi a vlastnosti pružnosti.[9]



Obr. 13 Lišta pohybové studie.

### 3.5 Identifikace systémů

Identifikaci systémů je možno rozdělit do dvou základních skupin. První tzv. Analytická identifikace, která spočívá v sestavování matematických modelů pomocí matematicko-fyzikálního modelování. Tato identifikace dostala název pouze Modelování. Do druhé skupiny patří Experimentální identifikace, která je v dnešní době spíše nazývána Identifikace systémů. V praxi se nejčastěji oba dva přístupy identifikace kombinují, aby se dosáhlo co nejpresnějších výsledků. Každý přístup má své klady i zápory. Hlavní odlišností metody Modelování je schopnost analyzování dynamických vlastností zařízení ještě před skutečným vyrobením. U této metody není potřeba mít fyzický reálný model. Zde jsou výhody v možnosti snadného dimenzování zařízení a další možné modifikace, což urychluje postupy k závěrečnému výrobku s minimalizací potřeby výroby prototypů.

Experimentální identifikace systémů znamená vyhodnocení dynamických vlastností systému a především stanovení matematického modelu systému za pomoci experimentálních metod. Na systém přivádíme vhodné testovací signály a následně sledujeme odezvu na signál. Po vyhodnocení odezvy signálu určíme model systému.

U metod experimentální identifikace systémů je zapotřebí mít skutečný reálný objekt pro možnosti identifikace. V dnešní době vznikají simulační programy, které tuto nevýhodu anulují. Zařízení v simulačním programu se chová jako reálný skutečný model. Není zde vyžadovaná dokonalá znalost struktury systému a popis vnitřních procesů, jsou však velké nároky na přístrojové vybavení. Čím širší a hlubší předem dané znalosti o identifikovaném objektu, tím snadněji je možné provést identifikaci, vyhodnotit získaná data a vytvořit přesnější matematický model.

Metoda modelování systémů požaduje hlubokou znalost fyzikálních dějů a matematického popisu těchto dějů pro vyjádření charakteristik ve zkoumaném systému. Potřebná znalost procesů je v porovnání s metodou identifikace systémů mnohem vyšší. Experimentální metody identifikace vyžadují nákladné přístrojové vybavení. **Chyba! Nenalezen zdroj odkazů.**

#### 3.5.1 Rozdělení identifikačních metod

Identifikační metody můžeme rozdělit podle typu matematických modelů a podle vstupního signálu.



## Druhy matematických modelů

Základní rozdělení matematických modelů jsou parametrické modely (parametric) a neparametrické modely (nonparametric).

### Parametrické modely

Parametrické modely popisují systémy, které jsou reprezentovány přenosem, diferenciálními nebo diferenčními rovnicemi v závislosti na diskrétním nebo spojitém modelu. Ve srovnání s neparametrickými modely poskytují přesnější odhad, pokud jsou známy předchozí vlastnosti systému. Z matematického hlediska zastupují rovnice, algebraické vztahy, které zahrnují explicitně vyjádřené parametry modelu.

### Neparametrické modely

Tyto modely nejčastěji reprezentují funkční závislost mezi daným vstupním signálem a k němu odpovídajícím výstupním signálem. Mnohdy jsou vyjádřeny graficky pomocí měření odezvy signálu na vstupní signál. Většinou vyjadřují váhovou funkci a přechodovou charakteristiku v grafu. Hodnoty parametrů modelu lze získat až vyhodnocením neparametrických modelů.

## Rozdělení vstupních signálů

Základní rozdělení vstupních signálů je na deterministické a stochastické. Signály deterministické můžeme analyticky vyjádřit. Mezi tyto signály můžeme zařadit skokovou změnu, rampový signál, puls, harmonický signál a další.

Stochastické, neboli náhodné, signály se odlišují od deterministických tím, že je nelze analyticky popsat. Každý tento signál je jedinečný a neopakovatelný. Hlavním představitelem stochastických signálů je Bílý šum.

Další kategorií v dělení signálů jsou pseudonáhodné signály. Jedná se o speciální druh stochastických signálů, kde jsou vlastnosti za daných podmínek stejné. Pseudonáhodné signály jsou generovány deterministicky. Příkazy pro generování pseudonáhodných signálů pro programovací prostředí Matlab mohou být například `rand`, `randn`, `awgn`. [7]

### 3.5.2 Postup pro identifikaci

Identifikace se skládá z několika základních kroků, které je nutno dodržovat:

1. Získání dat pro identifikaci a ověření.
2. Vlastní identifikace.
3. Ověření výsledného modelu.
4. Zobrazení a uložení modelu.

Nejdříve musíme získat data potřebná pro identifikaci, abychom mohli odhadnout model pro regulovanou soustavu. Dále získaná data můžeme upravit pomocí různých filtrů, rozdělit, provést “detrend”, udělat normalizace nebo obnovit chybějící data a další. Po upravení dat se musíme rozhodnout, jaký druh odhadu modelu zvolíme. Do metod neparametrického odhadu modelu řadíme metodu pro odhad impulsní odezvy nebo frekvenční odezvy. Tyto metody jsou často méně přesné než další metody nazvané jako Parametrický

odhad modelu. Parametrický odhad modelu poskytuje již komplexnější náhled do systému. Pokud známe nějaké informace o modelu, můžeme použít metodu pro částečně známý odhad modelu. Další možností je použití metody rekurzivního odhadu modelu. Po získání odhadu modelu můžeme provést analýzu modelu a zjistit například informace o frekvenční charakteristice nebo určit přesnost modelu popisující reálné chování regulované soustavy.

### 3.5.3 Metoda odhadu parametrického modelu

Existuje celá řada metod pro odhad parametrického modelu. Zmíníme čtyři základní kategorie, které je také možno vybrat v grafickém programovacím prostředí LabVIEW:

- Polynomiální lineární model (General-Linear Polynomial Model).
- Přenosový model (Transfer Function Model).
- Model nul a pólů (Zero-Pole-Gain Model).
- Stavový prostorový model (State-Space Model).
- Uživatelsky definovaný model (User-Defined Model).

Pro potřeby naší práce byla vybrána kategorie Polynomiální lineární model.

#### Polynomiální lineární model

Tento model je vhodný pouze pro diskrétní systémy. Diskrétní systém můžeme popsat podle následující rovnice (rce.3.4).

$$y(k) = z^{-n}G(z^{-1}, \theta)u(k) + H(z^{-1}, \theta)e(k) \quad (3.4)$$

$u(k)$  .....vstup do systému.

$y(k)$  .....výstup ze systému.

$e(k)$  .....regulační odchylka (chyba výstupu).

$G(z^{-1}, \theta)$  .....deterministický přenos systému.

$H(z^{-1}, \theta)$  .....stochastický přenos systému.

Deterministický přenos určuje vztah mezi vstupním a výstupním signálem. Stochastický přenos zjišťuje, jak náhodné regulační odchylky ovlivňují výstupní signál. Symbol  $z^{-1}$  je definován v následující rovnici:

$$z^{-1}x(k) = x(k-1) \quad (3.5)$$

$$z^{-2}x(k) = x(k-2) \quad (3.6)$$

... ..

$$z^{-n}x(k) = x(k-n) \quad (3.7)$$

$z^{-n}$  .....definuje dobu zpoždění mezi vstupem a výstupem.

$G(z^{-1}, \theta)$  a  $H(z^{-1}, \theta)$  jsou polynomy definované následující rovnicí:

$$G(z^{-1}, \theta) = \frac{B(z, \theta)}{A(z, \theta)F(z, \theta)} \quad (3.8)$$

$$H(z^{-1}, \theta) = \frac{C(z, \theta)}{A(z, \theta)D(z, \theta)} \quad (3.9)$$

$\theta$  ..... je vektor popisující parametry modelu.

V následující rovnici je pro zjednodušení a lepší orientaci vynechán symbol  $\theta$ . Rovnice (rce.3.10), (rce.3.11), (rce.3.12), (rce.3.13), (rce.3.14) definují polynomy  $A(z)$ ,  $B(z)$ ,  $C(z)$ ,  $D(z)$  a  $F(z)$ :

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{k_a} z^{-k_a} \quad (3.10)$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{k_b-1} z^{-(k_b-1)} \quad (3.11)$$

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{k_c} z^{-k_c} \quad (3.12)$$

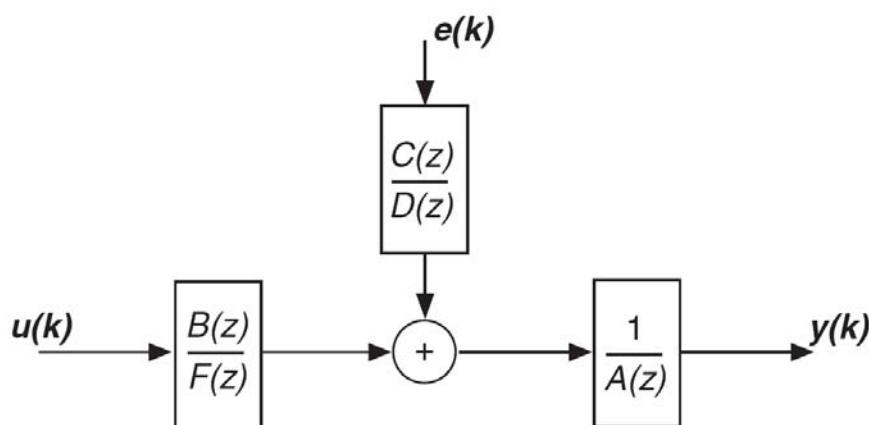
$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{k_d} z^{-k_d} \quad (3.13)$$

$$F(z) = 1 + f_1 z^{-1} + f_2 z^{-2} + \dots + f_{k_f} z^{-k_f} \quad (3.14)$$

$k_a, k_b, k_c, k_d, k_f$  ..... řád modelu.

Následující rovnice (rce.3.15) popisuje Polynomiální lineární model:

$$A(z)y(k) = \frac{z^{-n}B(z)}{F(z)}u(k) + \frac{C(z)}{D(z)}e(k) = \frac{B(z)}{F(z)}u(k-n) + \frac{C(z)}{D(z)}e(k) \quad (3.15)$$



Obr. 14 Struktura Polynomiálního lineárního modelu.

Polynomiální lineární model poskytuje flexibilitu jak pro deterministický přenos systému, tak také pro stochastický přenos systému. Modifikací každého z polynomů  $A(z)$ ,

$B(z)$ ,  $C(z)$ ,  $D(z)$  nebo  $F(z)$  na jedničku vzniknou jednodušší modely jako například AR, ARX ARMAX, Output-Error, Box-Jenkins model.

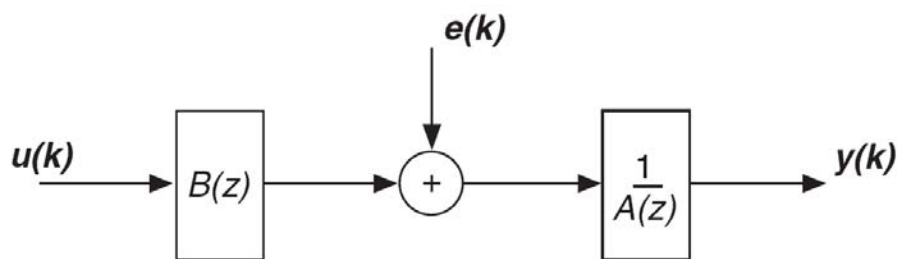
Dále pro potřeby naší práce vybíráme model ARX.

### ARX model

ARX model, Autoregresní model s vnějšími podmínkami (Autoregressive with exogenous terms), Tento model je zjednodušen od Polynomiálního lineárního modelu tak, že polynomy  $C(z)$ ,  $D(z)$  a  $F(z)$  jsou rovny nule. Následující rovnice (rce.3.16) popisuje ARX model:

$$A(z)y(k) = z^{-n}B(z)u(k) + e(k) = B(z)u(k-n) + e(k) \quad (3.16)$$

$$z^{-n}u(k) = u(k-n) \quad (3.17)$$



Obr. 15 Struktura modelu ARX.

Patří mezi nejjednodušší modely, které obsahují vstupní signál tzv. Stimulus. V tomto modelu má deterministický a stochastický přenos systému stejně nastavené póly. Jestliže není použit Bílý šum pro regulační odchylku  $e(k)$ , spojení mezi deterministickým a stochastickým systémem může být odhadnuto za pomoci ARX modelu. Při nastavení vyššího řádu modelu je minimalizována chyba odhadu. Při zvýšení řádu modelu se však mění dynamické vlastnosti a také stabilita modelu.

Pro ARX model se používá identifikační metoda nazvaná Metoda nejmenších čtverců, což je speciální případ metody chyby predikce. Metoda nejmenších čtverců je jedna z neúčinnějších metod polynomiálního odhadu, protože řeší lineární regresivní rovnice v analytickém tvaru.[14]

### 3.5.4 Metoda odhadu částečně známého modelu

Metod odhadu částečně známého modelu se používá, jestliže jsme schopni matematicky alespoň částečně popsat soustavu. Předpokládáme, že všechny parametry modelu jsou nastavitelné. Nastavení parametrů popisující reálný svět mohou mít různé omezení. Například v některých chemických procesech musí být průtok kapaliny pouze v jednom směru nebo při popisu dynamiky nemůže být například hmotnost záporná atd. Proto je u této metody možnost nastavení omezení parametrů. Možnosti nastavení omezení jsou maximální, minimální a počáteční hodnota.

### **Srovnání s neznámým a částečně známým modelem**

V praxi je většina modelů částečně známá, protože známe dynamiku systému nebo některé fyzikální parametry. Neznámý model, který se identifikuje za pomoci metody odhadu neparametrického nebo parametrického modelu, se také označuje jako černá skříňka (black-box) a částečně známý model jako šedá skříňka (grey-box).

Metoda odhadu neznámého systému nevyužívá znalosti o systému. Nenastavují se omezení parametrů, můžeme pouze nastavit řád modelu. Naopak metoda odhadu částečně známého modelu využívá znalosti o systému a je tudíž možnost nastavení omezení parametrů. Omezení reflektují znalosti o fyzickém systému. S omezeními je model určen přesněji a také zvyšuje možnost identifikace. [14]



## 4 Realizace modelu v NI SoftMotion Module for SolidWorks

### 4.1 Vlastnosti NI SoftMotion Module for SolidWorks

Spolupráce společností National Instruments a Dassault Systemes pomáhá pro řešení virtuálního prototypování (Rapid Prototyping). National Instruments je společnost zabývající se návrhem řízení, výrobou automatických zkušebních zařízení a dalších. Společnost Dassault Systemes SolidWorks Corp. je přední firmou v oblasti konstrukčního navrhování. Spolupráci těchto firem vznikl nástroj pro vývoj mechatroniky, pomáhající strojírenským technikům a řídicím technikům snížit náklady a omezit rizika při konstrukci strojů. Nástroj umožňuje propojení virtuálního CAD modelu, který je vytvořen v programu SolidWorks do prostředí NI Labview. Propojením můžeme vyžít všechny funkce a nástroje, které jsou v Labview. Integrace je ideální nástroj pro tvorbu digitálních virtuálních prototypů, pomáhající technikům a vědcům navrhovat, optimalizovat, vyhodnocovat a vizualizovat funkci strojů ještě před sestavením fyzických prototypů. Virtuální prototyp znázorňuje nejen vzhled, ale i pohyb nového stroje.

Integrací LabVIEW 2009 NI SoftMotion Module a SolidWorks 2009 vzniká vývojové prostředí, které je ideální pro virtuální prototypování. Propojením vytvořeného CAD modelu do LabView se automaticky přenesou všechny vlastnosti definované v programu SolidWorks, jako jsou určení motorového pohonu a snímače polohy nebo hmotnost sestavy, třecí účinky, vazby v sestavě.

Možnosti, které poskytuje nástroj pro propojení programů LabVIEW 2009 NI SoftMotion Module a SolidWorks, jsou následující:

#### **Návrh trajektorie pohybu**

Z prostředí NI LabVIEW je možno ovládat kinematickou studii, která obsahuje sekvenci pohybů pro dané díly sestavy v SolidWorks. Zatím je možnost realizace pohybu ve dvou dimenzích tzn. přímý, konturový, obloukový pohyb a také složité pohyby pomocí elektronických převodů a vaček. Po zadání posunu nebo natočení daného prvku získáme rychlost a zrychlení.

#### **Vizualizace**

Prostřednictvím NI LabVIEW je možno realizovat vizualizace v prostředí SolidWorks. Vizualizací můžeme zjistit pohyby jednotlivých dílů mechanismu a tím zkontrolovat jejich správnost a proveditelnost bez nutnosti vytvoření fyzického prototypu.

#### **Detekce kolizí**

SolidWorks v nadstavbě SolidWorks Motion disponuje nástrojem pro detekci kolizí u pohybové studie. Z toho plyne využití nástroje při propojení SolidWorks a NI LabVIEW u pohybové simulace. Můžeme tak předejít kolizím u skutečného reálného modelu a to úpravou trajektorie pohybu změnou, například momentu v soustavě nebo konstrukční změnou dílu sestavy.

### **Doba pohybové studie**

Využitím vypočítané pohybové studie systému, která obsahuje reálné fyzikální vlastnosti, jako jsou dynamické parametry, hmotnost, tření a další můžeme přesně vypočítat potřebnou dobu cyklu. Tím můžeme následně provést optimalizaci modelu.

### **Návrh parametrů pohonu**

Po výpočtu simulace lze získat potřebné zrychlení jednotlivých akčních členů soustavy v programu SolidWorks. Před výpočtem musí být zadány dynamické parametry modelu soustavy a určeny požadavky na pohyb soustavy.

### **Řízení průběhu simulace mechatronického systému**

NI LabVIEW ovládá simulaci modelu v programu SolidWorks. Prostřednictvím NI LabVIEW můžeme spouštět a získávat data z modelu. V dnešní době toto propojení programů neumožňuje celkový návrh a simulaci řízení mechatronické soustavy. Stále zde chybí podstatné získávání parametrů z modelu. Chybí například možnost ovládat síly a momenty působící na soustavu. Můžeme ovládat pouze prvky jako je „motor“ u kterých můžeme určit, zda se bude jednat o pohyb přímočarý nebo rotační a zadávat mu parametry jako pozice, rychlost, zrychlení. [6]

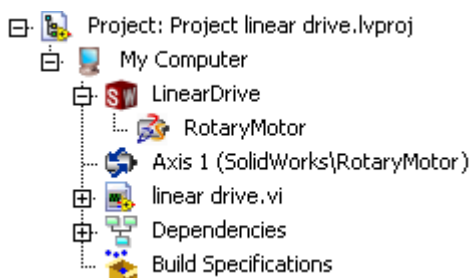
Pro korektní komunikaci mezi NI LabVIEW Softmotion a SolidWorks je zapotřebí program ve verzi LabVIEW 2009 nebo novější, LabVIEW NI SoftMotion Module Standard 2009 nebo novější a dále SolidWorks 2009 Service Pack 2.1 nebo novější a SolidWorks Motion.

## **4.2 Postup návrhu SoftMotion module**

Pro připojení pohybové studie z programu SolidWorks do LabVIEW je nejdříve zapotřebí nastavit na daný díl akční člen (motor), který uskutečňuje pohyb v sestavě. NI LabVIEW umožňuje snadné načtení sestavy z SolidWorks. Při větším počtu pohybových studií v SolidWorks je možnost výběru dané studie. LabVIEW automaticky načte všechny dostupné „motory“ v pohybové studii. Dále je zapotřebí načíst motory jako pohyblivé simulované osy. Na (obr. 16) je znázorněno načtení sestavy pod názvem LinearDrive, která obsahuje pouze jeden motor s názvem RotaryMotor. Dále je motor načten jako simulovaná osa, která se bude následně využívat v projektu.

Tato filozofie simulovaných os má snahu zjednodušit přechod mezi řízením virtuálního modelu na řízení reálného systému. Po ověření simulace na virtuálním modelu lze jednoduše změnit simulované osy na reálný aktuátor a otestovat řízení na reálném systému.[6]





Obr.16 Načtení simulovaných os v NI LabVIEW SoftMotion.

### Synchronizace s pohybovou studií

Synchronizace mezi pohybovou studií SolidWorks a prostředí LabVIEW je přímo implementována do aplikace SoftMotion. Do grafického programovacího prostředí je umístěna časová smyčka, která má nastavený synchronizační zdroj pro výpočet pohybové studie (Scan Engine). Při spuštění programu běží smyčka se stejnou časovou frekvencí jako simulace v pohybové studii SolidWorks. Všechny funkční bloky, které jsou umístěny uvnitř časové smyčky, se provedou jednou za periodu (Scan period). NI SoftMotion funkční bloky se také často používají v hardwaru s real-time operačním systémem, kde vytvářejí aplikace pro řízení pohybu.[6]

U funkčních bloků NI SoftMotion používaných v operačním systému Windows při simulaci v programu SolidWorks není zaručena korektní synchronizace s NI Scan Engine, protože operační systém Windows je nedeterministický. Real-time operační systémy jsou deterministické tzn. je zde zaručena synchronizace s NI Scan Engine.[18]

### 4.3 Popis jednotlivých bloků NI SoftMotion module

NI SoftMotion Module pro SolidWorks používá NI SoftMotion funkční bloky s NI Scan Engine pro vytvoření návrhu pohybové simulace. Funkční bloky umožňují vytvářet aplikace pro řízení deterministického pohybu. Existují také bloky pro sledování jednotlivých prvků. Funkční bloky komunikující přes API mohou být používány také pro komunikaci s hardwarem.

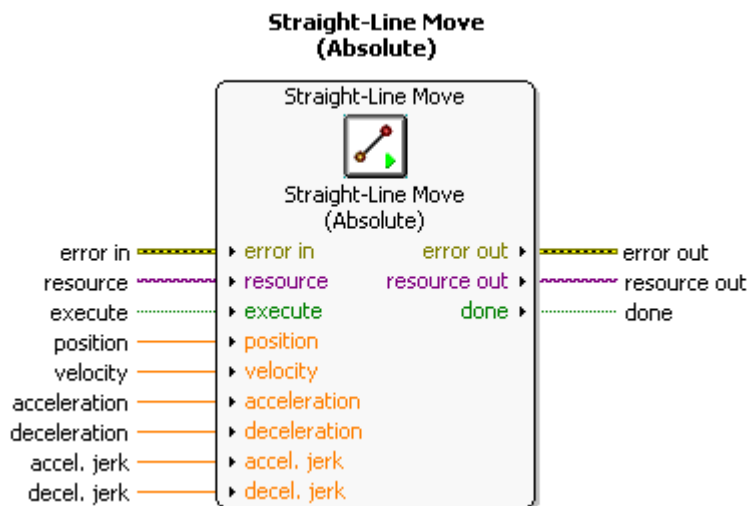
Popis jednotlivých základních funkčních bloků:

#### **Straight-Line Move (Přímý lineární pohyb)**

Tento funkční blok provádí přímý lineární pohyb za použití os nebo souřadného systému. Přímý pohyb spojuje dva body pomocí jedné nebo více os. Blok obsahuje základní vstupy, které jsou zadávání pozice, rychlosti, zrychlení, zpomalení, rychlost zrychlení (acceleration jerk), rychlost zpomalení (deceleration jerk). Chování pohybu můžeme změnit v závislosti na změny Straight-Line Mode (režimu). Funkční blok nabízí následující druhy režimů:

- Absolute (absolutní) – zadávání hodnot pozice v absolutních hodnotách.
- Relative (relativní) – zadávání hodnot pozice v relativních hodnotách.

- Velocity (rychlost) – u tohoto módu se nezadává pozice, ale jen rychlost, zrychlení atd.
- Relative To Capture – cílová pozice je závislá na předchozí pozici nebo na poslední nalezené indexové pozici.

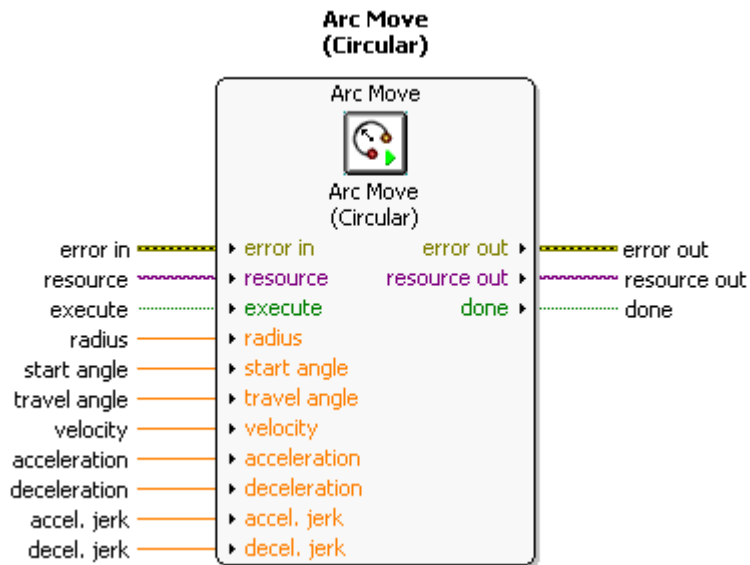


Obr.17 Funkční blok Přímého lineárního pohybu.

### Arc Move (Pohyb po oblouku)

Funkční blok Arc Move realizuje kruhový, sférický pohyb nebo pohyb po spirále. Pohyb po oblouku určují základní parametry poloměr, startovní úhel, úhel při pohybu. Další vlastnosti a možnosti pohybu nastavíme v určitém Arc Move Mode (režimu). Blok nabízí následující režimy:

- Circular (kruhový) – kruhový pohyb, který je definovaný v rovině  $x, y$  dvěma nebo třemi souřadnicemi.
- Spherical (sférický) – sférický pohyb, který je definovaný jako kruhový oblouk v rovině  $x, y$  transformován do souřadnic  $x, y$  a  $z$ . Prostorové souřadnice jsou definovány pomocí valení, odchylky a úhlu naklonění.
- Helical (spirálový) – pohyb po spirále je ve trojrozměrném prostoru, který se skládá z kruhu v rovině  $x, y$  a synchronizovaném lineárním dráze na ose  $z$ .

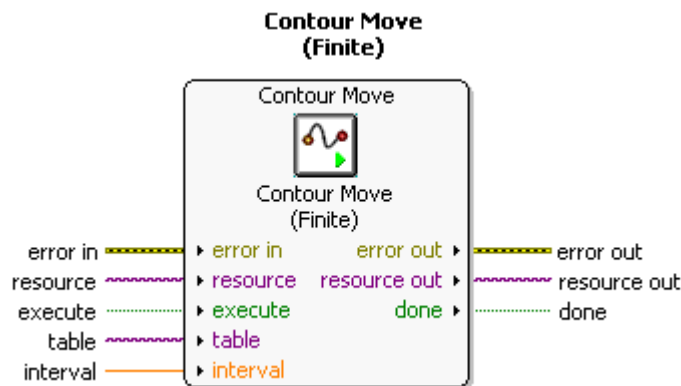


Obr. 18 Funkční blok Pohybu po oblouku.

### Contour Move (Pohyb po kontuře)

Poskytuje možnost pohybu po kontuře, která je daná sérií pozic. Software tyto body (pozice) extrapoluje na plynulou křivku. Pozice jsou uloženy v tabulce. Každý bod je zadáván absolutní pozicí. Typ pohybu po kontuře můžeme měnit v závislosti na režimu:

- Finite (konečný) – nemožnost aktualizace dat v tabulce během výpočtu.
- Repetitive (opakovaný) – stejně jak u předchozího nemožnost aktualizace dat. Je zde schopnost zadání počtu iterací získávání dat z tabulky.
- Continuous (plynulý) – data v tabulce mohou být dynamicky aktualizována.

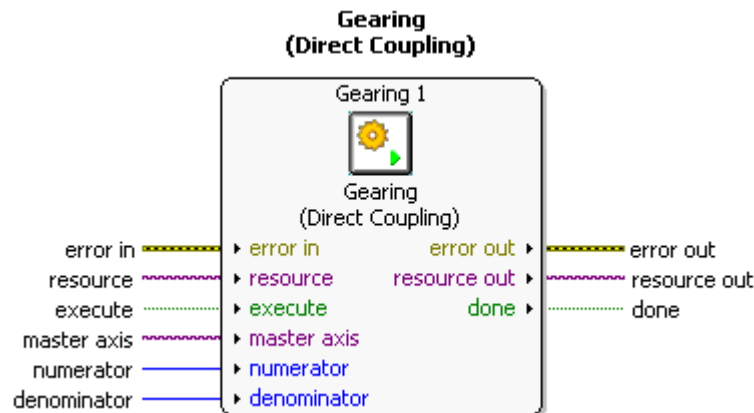


Obr.19 Funkční blok Pohybu po kontuře.

### Gearing (Převod)

Nastaví pro dané osy určitý převodový poměr. Převod je synchronizován na slave ose závislý na master zařízení, které může být kodér nebo trajektorie jiné osy. Převodový poměr nastavujeme pomocí numerator (čitatel) a denominator (jmenovatel). Změny nastavení převodu lze provádět v následujících režimech:

- Direct Coupling (Přímé spojení) – přímé spojení mezi master a slave osou. Rychlost slave osy je řízena přímo přes master osu.
- Gear Out – zrušení převodu mezi master a slave osou.

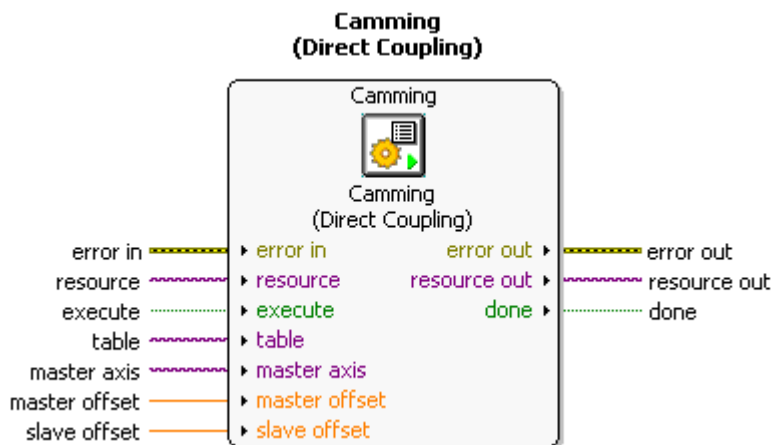


Obr.20 Funkční blok Převodu.

### Camming (Pohyb přes vačku)

Tento typ funkčního bloku se používá v aplikacích, kde jsou slave osy ovládány nelineárním profilem z master osy. Hodnoty jsou čteny z tabulky. Typy pohybu po vačce jsou závislé na následujících režimech:

- Direct Coupling (Přímé spojení) – při přímém spojení os master a slave nejsou brána v úvahu omezení. Rychlost slave osy je přímo řízena master osou.
- Cam Out – zrušení spojení mezi master a slave osou.



Obr.21 Funkční Blok Pohyb přes vačku.

### Read (Čtení)

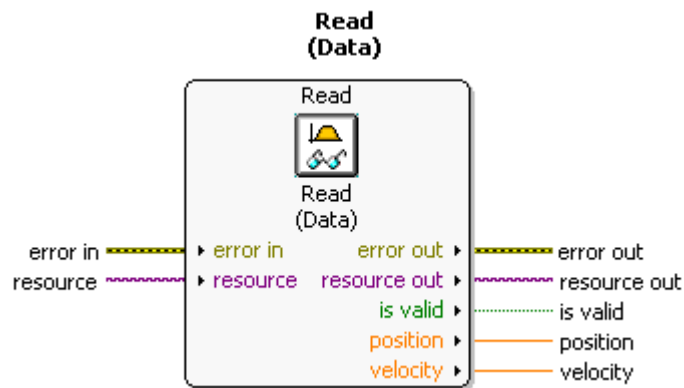
Čte stav a datové informace z os, souřadnice, zpětné vazby a dalších zdrojů. Jsou zde k dispozici různé metody pro čtení, které nám umožňují získávat informace z různých zdrojů. Následující metody pro čtení jsou k dispozici pro osy a souřadné systémy:

- Data – čte data pozice, rychlost, krouticí moment.

- Status (Stav) – určí stav z osy nebo souřadného systému.
- Trajectory Data (Data o trajektorii) – čte aktuální hodnoty trajektorie z osy nebo ze souřadného systému.

Následující metody jsou k dispozici pouze pro osy:

- Encoder (kodér) – čte informace z daného kodéru, který je připojen na ose.
- Motion I/O (Pohyb vstup/výstup) – čte stav a data ze vstupu a výstupu pohybu připojené k vybrané ose.
- Capture Data – čte poslední pozici na vybrané ose kodéru. Tato metoda není podporována pro propojení se SolidWorks.
- Digital Line – čte pouze stav digitálního vstupu. Zde se dá načíst senzor ze SolidWorks, ale je zde omezení, kde můžeme číst kritické hodnoty senzoru pouze typu boolean (T nebo F).
- Analog Line – čte stav analogového vstupu. Metoda určena pouze pro nevázané osy.
- Faults – čte NI SoftMotion chybové informace z určité osy.



Obr.22 Funkční blok pro čtení.

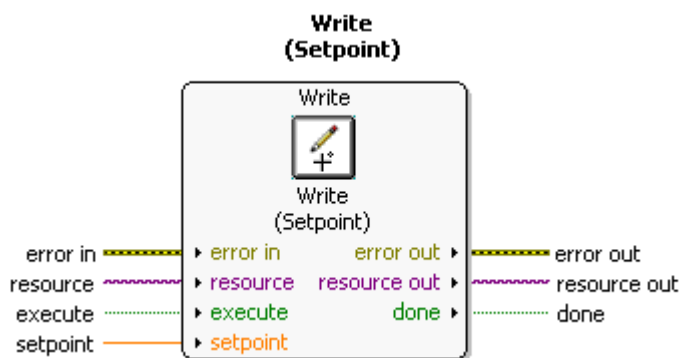
### Write (Zápis)

Následující metody pro zápis jsou určeny pro osy i souřadné systémy:

- Velocity Override (Překročení rychlosti) – zadání maximální mezní rychlosti.

Následující metody jsou určeny pouze pro osy:

- Setpoint (Zadaná hodnota) – načte nový bod pro osu.
- Digital Line – nastaví hodnotu digitálního výstupu. Tato metoda není podporována pro propojení se SolidWorks.
- Analog Line – nastaví hodnotu analogového výstupu. Metoda určena pouze pro nevázané osy.



Obr.23 Funkční blok pro zápis.

**Capture**

Zaznamenává pozici kodéru spojenou s externím vstupem, například stav snímače. Tento funkční blok není podporován pro propojení se SolidWorks. Můžeme vybrat z následujících metod:

- Single – jednoduché snímání.
- Disable – vypnutí snímání pozice.

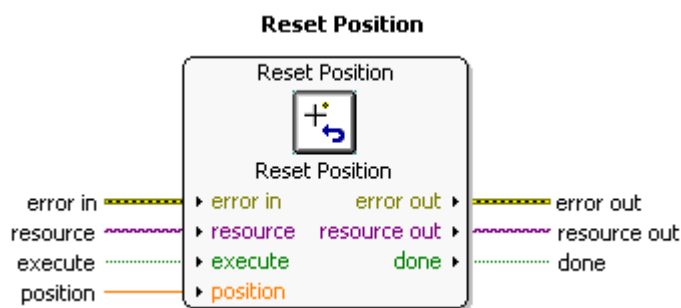
**Compare (Srovnání)**

Synchronizuje motor s vnějšími činnostmi a daným kódem pozic. Je-li dosaženo zadané hodnoty pozice, nastavitelný puls je přerušen. Chování srovnání pozice lze změnit následujícími režimy:

- Single Absolute (Absolutní)
- Single Relative (Relativní)
- Periodic Continuous (Nepřetržitý) – průběžně provádí porovnání.
- Disable (Vypnutý) – zakázání porovnání.

**Reset Position (Vynulování pozice)**

Vynulování pozice dané osy nebo souřadnice.

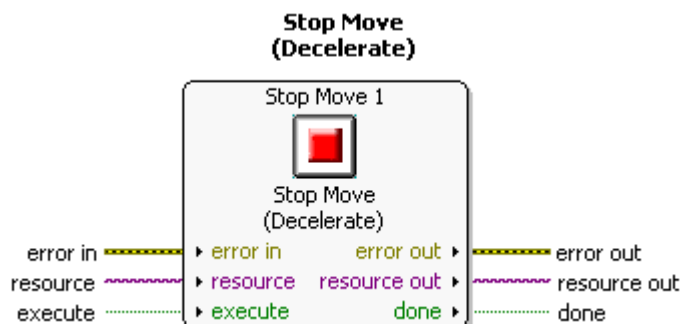


Obr.24 Funkční blok pro vynulování pozice.

**Stop (Zastavení)**

Zastaví současný pohyb na ose nebo na souřadnici. V závislosti na režimu lze měnit chování zastavení pohybu:

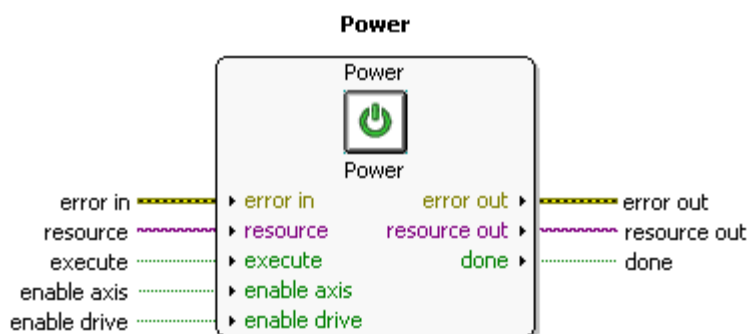
- Decelerate (Zpomalení) – začne snižovat rychlost daným zpomalováním. Cílové pozice nedosáhne.
- Immediate (Okamžité zastavení) – zastaví se okamžitě.
- Disable Drive (Zakázání) – vypnutí celé komunikace a řízení.



Obr.25 Funkční blok pro zastavení.

### Power (Vypnutí)

Zapíná nebo vypíná pohony na daných osách nebo souřadnicových systémech.



Obr.26 Funkční blok vypnutí.

### Clear Faults (Opravení chyb)

Opravuje chyby, upozornění a NI SoftMotion poruchy.



Obr.27 Funkční blok pro opravení chyb. [8]

## 4.4 Realizace virtuálního modelu Lineárního pohonu

### Lineární pohon, část Stewartovy platformy

Lineární pohon představuje jeden lineární akční člen, který je součástí výše zmíněné Stewartovy platformy. Platforma je složena ze šesti totožných lineárních pohonů (nohou).

Pro simulaci řízení lineárního pohonu je nejdříve zapotřebí vytvořit 3D virtuální model v prostředí SolidWorks a poté jej následně převést do pohybové simulace SolidWorks Motion. Z prostředí NI LabVIEW SoftMotion module bude prováděno řízení virtuálního a poté i reálného modelu.

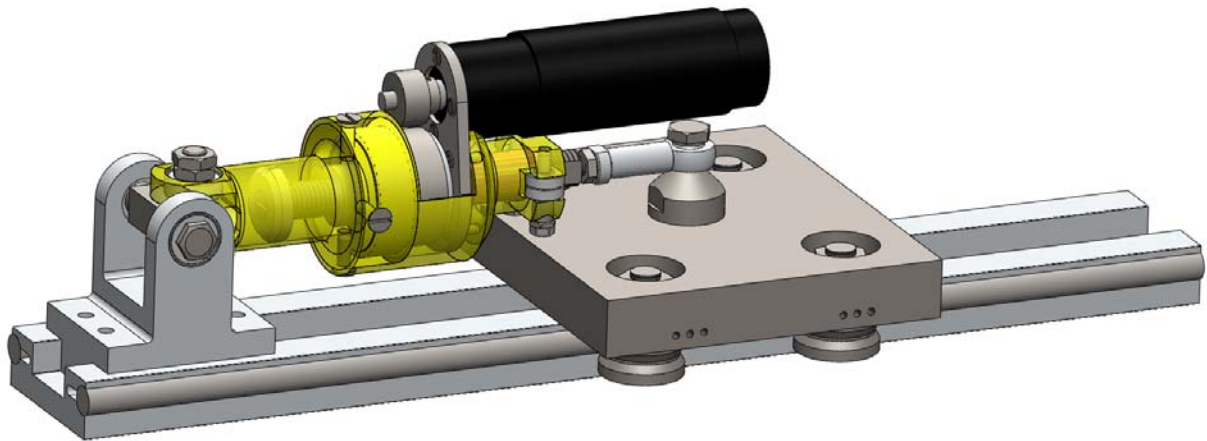
Hlavní funkční části lineárního pohonu jsou složeny ze čtyř základních dílů, které jsou motor, hlavní tělo, vozík a lineární vedení. Motor uskutečňující pohyb je spojen pomocí ozubeného převodu s ozubenou maticí v hlavním těle. Ozubená matice slouží také pro převod rotačního pohybu na lineární za pomoci kuličkového šroubu. Kuličkový šroub je spojen sférickou vazbou s vozíkem. Vozík koná pohyb po lineárním vedení za pomoci rolnic pohybujících se po tyči. Hlavní tělo obsahuje kuličkový šroub s ozubenou maticí. Tělo je zavazbeno pomocí kloubu k lineárnímu vedení.

Lineární pohon byl již navržen a vymodelován v 3D CAD aplikaci Autodesk Inventor. Model lineárního pohonu nelze převést z prostředí Autodesk Inventor do prostředí SolidWorks, aby byly zachovány veškeré vazby a aby se model choval korektně. Proto je nutné celý model přemodelovat v aplikaci SolidWorks 2009 nebo vyšší. Rozměry a definice materiálů jednotlivých dílů byly získávány z výkresů vytvořených v aplikaci Autodesk Inventor nebo byly vkládány normalizované součásti ze SolidWorks Toolbox, u nichž byly definovány katalogové hodnoty materiálu a rozměry.

Při skládání sestavy bylo zapotřebí brát v úvahu funkčnosti jednotlivých vazeb mezi díly, které se stanou po převedení do pohybové studie funkčními pohyblivými vazbami. Při dynamickém výpočtu v pohybové studii nastávaly problémy převazbení sestavy, proto sestava musela být vytvářena s velkým důrazem na skládání vazeb. Řešič v pohybové studii nedokázal vyřešit ozubený převod mezi ozubeným kolem spojujícím motor a ozubeným kolem obsahujícím matici pro kuličkový šroub. Tyto díly musely být zjednodušeny na prostý válec, u matice na díru a musely být spojeny speciální strojní vazbou převod a šroub. Dále měl řešič problémy s tangenciálními vazbami, které byly umístěny mezi vedením a rolnicemi na vozíku. Tyto vazby musely být také zjednodušeny přidáním dvou rovin, po kterých se rolnice posouvají.

Nakonec pro korektní a rychlé výpočty pohybové studie musela být sestava tvořena podsestavami pro zmenšení počtu vazeb v sestavě. Podsestavy se mohou řešit buď jako pevné (rigid), což znamená, že je celá podsestava chápána jako jeden díl, anebo jako flexibilní (flexible), podsestava je rozložena na jednotlivé díly a vazby jsou přeneseny do „celkové“ sestavy. Byly vytvořeny podsestavy, které byly řešeny jako pevné s největším možným počtem dílů, aby se snížil počet vazeb v sestavě. Na (obr. 28) je znázorněn výsledný virtuální model lineárního pohonu, který slouží pro účely propojení s NI LabVIEW a jeho řízení, které je popsáno v dalších kapitolách.



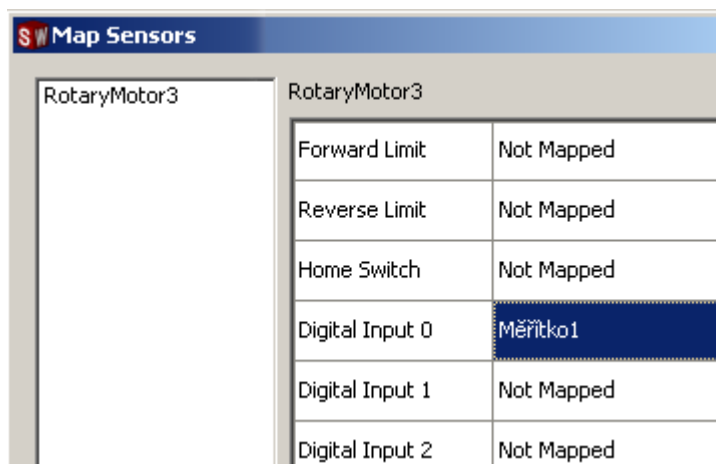


*Obr.28 Virtuální model lineárního pohonu v SolidWorks.*

#### **4.5 Připojení realizované pohybové studie do NI LabVIEW**

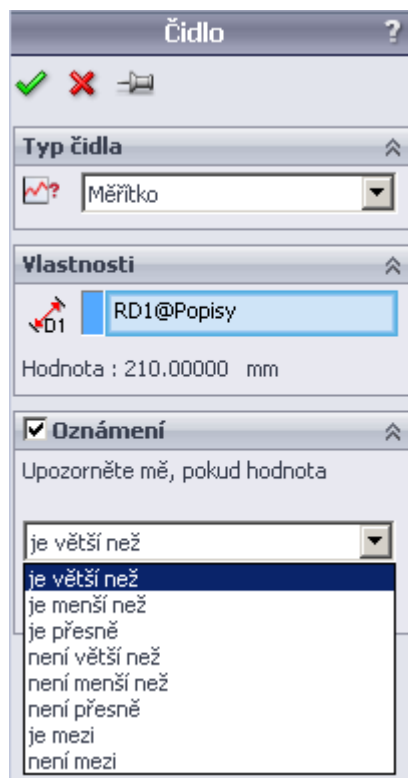
Pomocí dostupných nástrojů NI LabVIEW Softmotion module bylo provedeno připojení pohybové studie z programu SolidWorks do LabVIEW. Postupem času se tento nástroj stal pro další využití nepoužitelný. Pro ovládání motoru byl použit funkční blok ze Softmotion module pod názvem Straight-Line Move (Přímý lineární pohyb). Je možnost nastavení pozice, rychlosti, zrychlení atd. Nevýhodou tohoto bloku je, že není možnost zadání momentu.

Dále bylo zapotřebí získávat hodnoty pozice a rychlosti přímo na dílu vozíku, který se lineárně pohybuje po rolnicích. Pro čtení dat je určen funkční blok Read, který čte pouze pozice a rychlosti jen z dílu, na kterém je připojen motor. To znamená, že není možnost číst data až na vozíku. Alternativou je umístění senzoru v SolidWorks, který by umožnil čtení dat. Jediný funkční blok, který je vhodný na čtení dat ze senzoru je blok Read s režimem Digital Line. Pro korektní komunikaci musí být senzor v SolidWorks pouze typu „měřítko“ (Measurement senzor). Senzor je mapován do prostředí NI LabVIEW SoftMotion module.



Obr.29 Mapování senzorů.

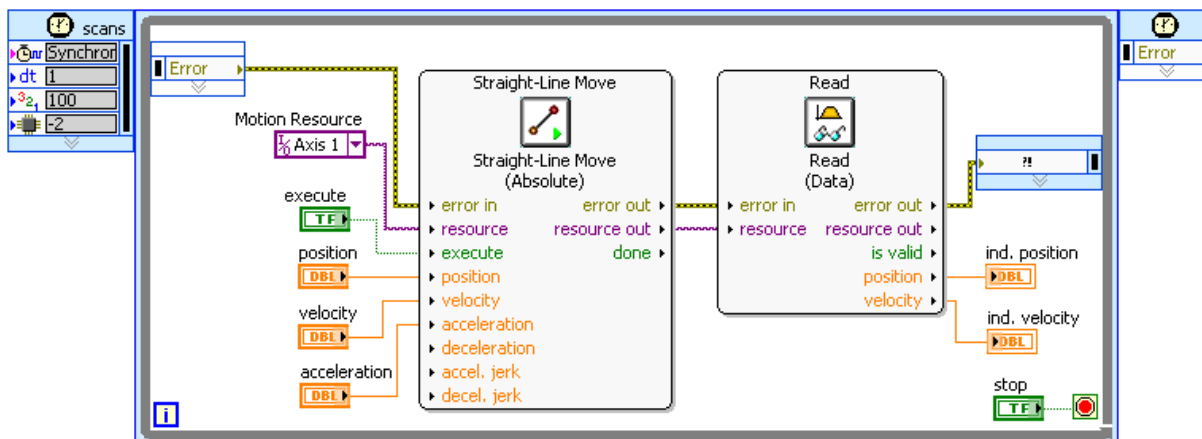
Zde nastává problém. Funkční blok Read s režimem Digital line, umožňuje číst pouze kritické hranice senzoru a to pouze v případě, je-li senzor o určitém rozměru True nebo False. Kritické meze daného senzoru se nastavují přímo v prostředí SolidWorks (obr. 30).



Obr.30 Nastavení kritických mezí senzoru v programu SolidWorks.

Aby byly získány požadované hodnoty, které budou důležité pro další úkol diplomové práce a to identifikaci systému, musíme zvolit jiný postup. Využijeme předchozí řešení v diplomové práci Ing. Fridrichovského [6], který pracoval se starší verzí propojení SolidWorks a NI LabVIEW pod názvem NI LabVIEW-SolidWorks Mechatronics Toolkit.

Navrhl nový toolkit pro propojení programů přes API, který nám zjistí souřadnice daného dílu v sestavě SolidWorks. Za pomoci tohoto členu je již možné získat potřebné informace o daném dílu jako je pozice, rychlost, zrychlení, které budou potřeba v následující kapitole pod názvem identifikace lineárního pohonu.



Obr.31 Schéma zapojení pomocí NI Softmotion module.

## 4.6 Zhodnocení výhody-nevýhody

### Výhody

Propojení nástrojů SolidWorks a NI LabVIEW prostřednictvím NI SoftMotion Module for SolidWorks poskytuje množství výhod. Hlavní výhodou je uživatelsky jednoduché prostředí, které umožňuje snadné připojení 3D CAD soustavy do prostředí LabVIEW, namapování pohonů (os) ovládajících soustavu. Další velmi podstatná výhoda je časová synchronizace s výpočtovou pohybovou studií, nastavení koncových dorazů a další. Soustavu lze řídit pomocí kinematických veličin. Je zde implementován PID regulátor určený pro kinematické řízení. Tímto nástrojem lze simulovat řízení virtuální soustavy a následně toto řízení využít pro reálnou soustavu. Virtuální osy lze nahradit za reálné motory použitím NI-cRIO.

### Nevýhody

Propojení však přináší také nevýhody, které byly při diplomové práci nalezeny. Pohony (osy) nelze ovládat pomocí dynamických veličin, jako například momentem, silou. Nelze získávat kinematické a dynamické veličiny ze senzorů. Umožňuje pouze získávat data z namapovaného pohonu. Toto propojení není vhodné pro dynamické řízení soustavy.



## 5 Modelování a simulace pomocí NI LabVIEW a SolidWork API

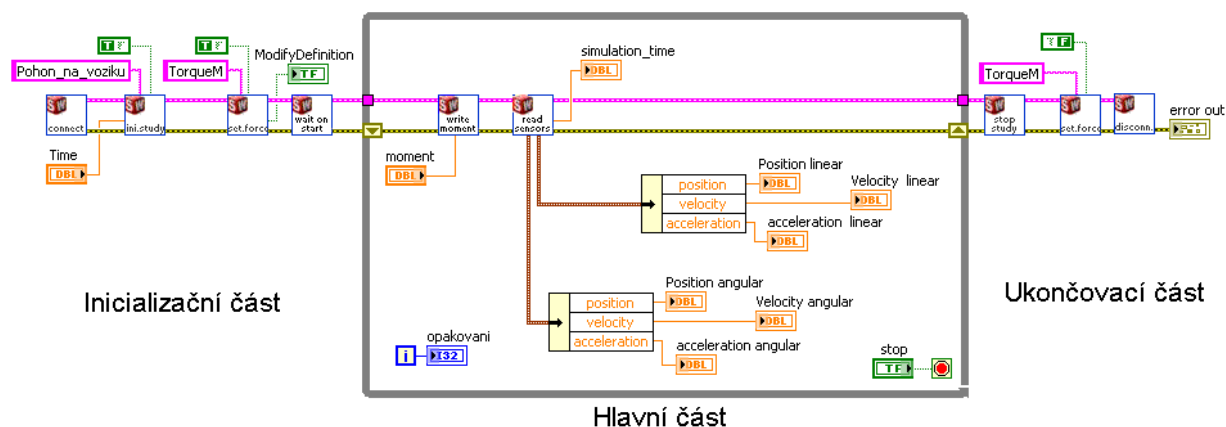
Na základě nedostatků propojení SolidWorks a LabVIEW pomocí nástroje NI LabVIEW SoftMotion for SolidWorks bylo nutné od této koncepce ustoupit. Proto bylo použito propojení, které naprogramoval Ing. Fridrichovský[6] ve své diplomové práci. Pro jednoduchost budeme toto propojení nazývat „toolkit“. Tento toolkit vznikl ještě před uvedením Softmotion module na trh. Toolkit ovládá SolidWorks přes aplikační programové prostředí (API), které je zveřejněno v SolidWorks API SDK.

### 5.1 Připojení pomocí toolkitu

Základní struktura toolkitu byla využita pro tuto práci. Některé bloky z toolkitu byly modifikovány a použity. Toolkit se skládá z bloků, které plní posloupnost operací, které jsou potřeba pro propojení LabVIEW a SolidWorks. Posloupnost těchto operací je následující:

- Připojení k SolidWorks (Inicializace referencí),
- inicializace pohybové studie,
- inicializace silového působení (moment),
- synchronizace spojení,
- zápis momentu,
- čtení ze senzorů,
- zastavení pohybové studie,
- ukončení silového působení,
- ukončení spojení s aplikací SolidWorks.

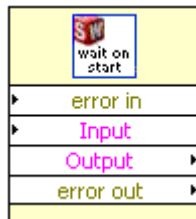
Zapojení s použitým toolkitem je zobrazeno na (obr. 32).



Obr.32 Schéma zapojení s novým toolkitem.

Podrobné popsání jednotlivých bloků tvořících toolkit od Ing. Fridrichovského je vysvětleno v jeho diplomové práci[6]. Nový blok přidáný do toolkitu odstraňuje problémy s počáteční synchronizací. Blok se nazývá Synchronizace spojení (obr. 33), který čeká, až je poprvé zavolán callback. Což znamená, dokud není spuštěn simulační výpočet pohybové

studie v SolidWorks, tak se z fáze inicializační nepřejde do hlavní fáze programu. Je-li výpočet pohybové studie v SolidWorks spuštěn, běh programu postupuje dále do hlavní části.



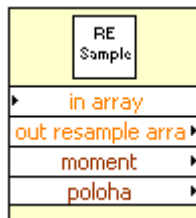
Obr. 33 Synchronizace spojení.

Tímto toolkitem byly odstraněny veškeré nevýhody předchozího propojení. Motor není stimulován kinematickými veličinami, jako jsou poloha, rychlost, zrychlení atd., ale je ovládán momentem nebo silou. Také je možno získávat hodnoty pozice, rychlosti a zrychlení na vozíku, což je pro řešení další úlohy Identifikace systému nezbytné.

## 5.2 Identifikace lineárního pohonu v prostředí LabVIEW

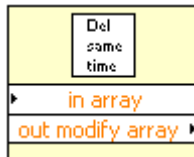
Pro získání matematického popisu dynamiky simulované soustavy je zapotřebí provést tzv. identifikaci systému. Matematický popis je důležitý pro návrh řízení, který je dalším cílem práce. Na soustavu přivádíme vhodné stimulační signály a sledujeme jejich odezvu. V našem případě stimulační signál představuje moment na motoru, který vytváří pohyb v soustavě. Odezvu sledujeme jako polohu, rychlost vozíku na lineárním vedení. Pro nej přesnější matematický popis získaný z identifikace je potřeba soustavu stimulovat reprezentativními hodnotami momentu. Byly použity tři různé druhy signálů. Jako první byl na soustavu přiveden (jednotkový) skok, dále moment se sinusovým průběhem a jako poslední byla soustava buzena náhodným momentem. Získaná data ze senzoru a hodnoty momentů byly použity pro identifikaci systému.

Při získávání dat bylo zjištěno nerovnoměrné vzorkování. Vzorkování dat získané ze simulační pohybové studie je dané složitostí výpočtu simulace. U výpočtu pohybové studie (simulace) se ukázalo, že SolidWorks není schopen vypočítávat data v konstantním časovém intervalu. Z toho vyplývá, že data jsou nerovnoměrně vzorkována. Pro využití identifikačních nástrojů je nutno vkládat data rovnoměrně vzorkovaná. Proto bylo potřeba udělat převzorkování (obr. 34).



Obr. 34 Převzorkování.

Data získaná ze simulace v SolidWorks měla stejné hodnoty času. Tato vlastnost řešiče výpočtu by byla problematická při identifikaci, proto data se stejnou hodnotou času musela být odstraněna. Blok, který tato data odstraní je zobrazen na (obr. 35).



Obr. 35 Odstranění dat se stejnými časovými hodnotami.

Pro identifikaci systému je možno vybrat z několika metod. Byla vybrána metoda odhadu částečně známého modelu.

### 5.2.1 Použití metody odhadu částečně známého modelu

Tato metoda využívá znalosti o systému. Metoda byla použita s pohybovou rovnicí odvozenou pomocí druhého Newtonova pohybového zákona tzv. rovnice kmitání (rce.5.1).

$$m\ddot{x} + b\dot{x} + kx = f(t) \quad (5.1)$$

$x$ .....zobecněná souřadnice.

$m$ .....zobecněná hmotnost.

$b$ .....zobecněné tlumení.

$k$ .....zobecněná tuhost.

$f(t)$ .....zobecněné buzení.

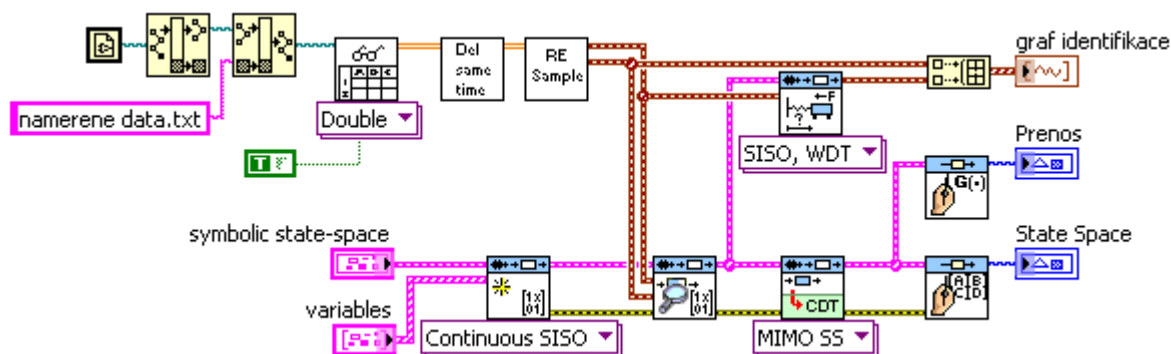
Převod diferenciální rovnice (rce.5.1) na stavové rovnice je zobrazen v (rce.5.2).

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f(t) \quad (5.2)$$

Stavové rovnice vyjádřené ve stavových maticích (rce.5.3).

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \quad C = [1 \quad 0] \quad D = [0] \quad (5.3)$$

Na (obr. 36) je znázorněno zapojení v grafickém programovacím prostředí LabVIEW pro identifikaci s použitou metodou odhadu částečně známého modelu. Pro metodu odhadu částečně známého modelu nejdříve musí vytvořit model. V našem případě podle rovnice kmitání. Dalším důležitým krokem je správné zvolení počáteční a hraniční podmínky. Po identifikaci systému lze získat stavový popis (State Space), přenos a další. Model systému lze uložit a následně s ním pracovat.



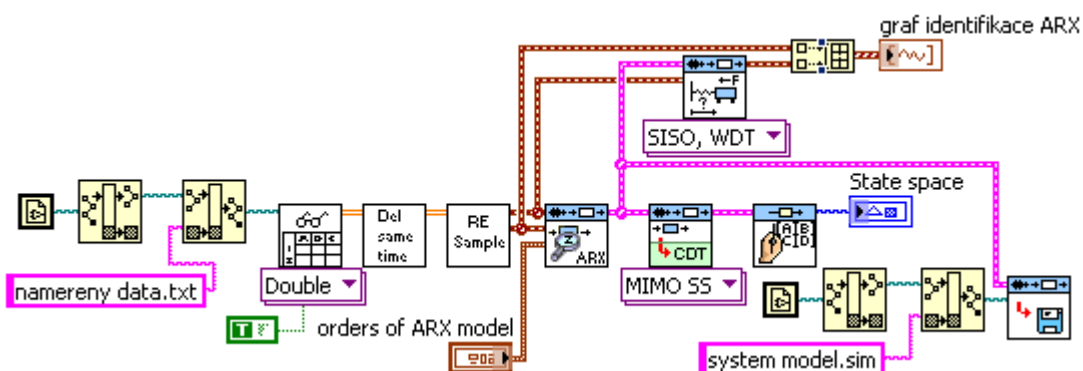
Obr. 36 Identifikace systému pomocí metody částečně známého modelu.

Při použití této metody bylo zjištěno, že velmi záleží na určení počátečních podmínek a také na minimální a maximální hodnotě hledaných parametrů. Při malé změně některé z podmínek byl výsledný matematický popis výrazně odlišný. Proto se od této metody odhadu částečně známého modelu ustoupilo a přešlo se k použití metody ARX.

## 5.2.2 Použití metody ARX

Metoda ARX (Autoregressive with exogenous terms) v překladu Autoregresní model s vnějšími podmínkami, patří do kategorie pod názvem Polynomiální lineární model. Metoda ARX byla využita pro identifikaci systému. V grafickém programovacím prostředí LabVIEW je tato metoda již implementována v nástavbě System Identification Toolkit. Na funkční blok stačí přivést odpovídající stimulační data a odezvu na tyto data. Dále můžeme získat přenos systému, stavový popis a další. Identifikovaný model systému lze uložit a dále s ním pracovat. Byl získán stavový popis popisující polohu a rychlost vozíku na lineárním vedení.

Tento model bude potřebný pro návrh řízení, kterým se zabývá další podkapitola. Na (obr. 37) je znázorněná identifikace použitím metody ARX v LabVIEW.



Obr. 37 Identifikace systému pomocí ARX.

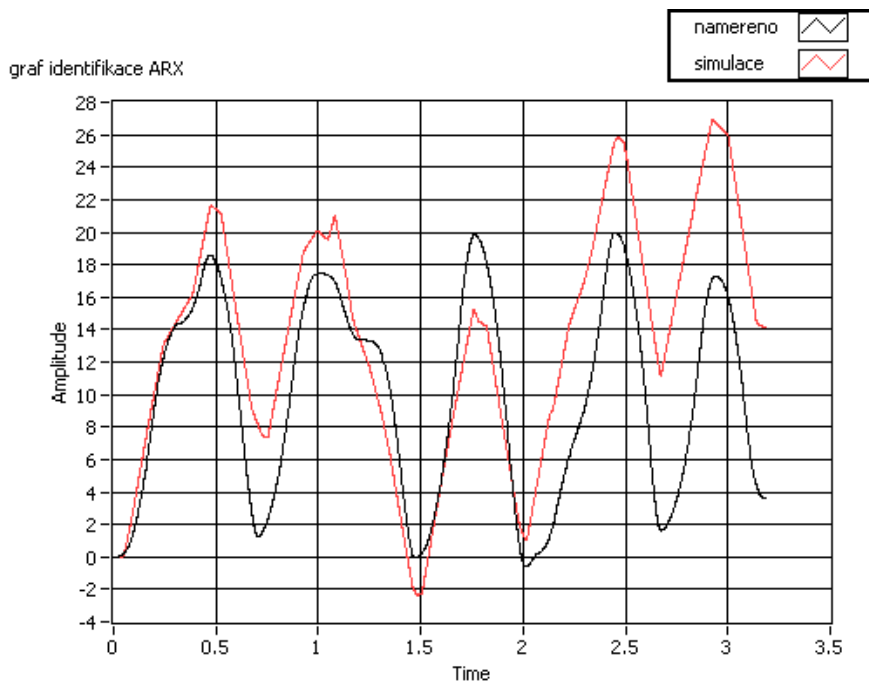
Pro identifikaci byl zvolen řád modelu ARX následovně:  $A = 2$ ,  $B = 2$  a zpoždění rovno 0. Získaný stavový popis z identifikace systému je vyjádřen v (rce.5.4).



$$x(k+1) = \begin{bmatrix} 1.75583 & 1 \\ -0.756014 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 3.67783E-4 \\ -3.05159E-5 \end{bmatrix} u(k) \quad (5.4)$$

$$y(k) = [1 \quad 0]x(k) + [4.03643E-5]u(k)$$

Na (obr. 38) je grafické znázornění odchylky naměřených hodnot a hodnot simulace při použití modelu získaného z identifikace. Bylo usouzeno, že tato odchylka je pro potřeby návrhu řízení přijatelná, proto je tento získaný matematický model použitelný pro další potřeby návrhu řízení.



Obr. 38 Porovnání simulace z identifikace s naměřenými hodnoty.

### 5.3 Návrh řízení lineárního pohonu

Pro návrh řízení byl použit stavový popis získaný z identifikace systému a také byl vytvořen stavový popis motoru pro potřeby řízení na reálném modelu.

Diferenciální rovnice stejnosměrného motoru jsou vyjádřeny (rce.5.5).

$$\begin{aligned} \dot{\phi} &= \omega \\ \dot{\omega} &= \frac{C_e}{J} i \\ i &= \frac{1}{L} (u - Ri - C_e \omega) \end{aligned} \quad (5.6)$$

$C_e$ .....konstanta motoru.

$L$ .....indukčnost.

$R$ .....odpor vinutí.

$J$ .....moment setrvačnosti.

$u$ .....napětí kotvy.

$\varphi$ .....úhlové natočení.

$\omega$ .....úhlová rychlost.

$i$ .....proud.

Z katalogu firmy Maxon[23] byly použity parametry DC motoru typu RE 35, které jsou uvedeny v tabulce níže:

Název parametru	Hodnota
Jmenovité napětí	$u = 42V$
Odpor vinutí	$R = 2.07\Omega$
Indukčnost	$L = 0.62mH$
Motorová konstanta	$C_e = 182rpm / V$
Moment setrvačnost	$J = 7.2e - 5kgm^2$

Tab. 1 Parametry DC motoru RE 35.

Stavové rovnice stejnosměrného motoru (rce.5.8). [19]

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\omega} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{C_e}{J} \\ 0 & -\frac{C_e}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \varphi \\ \omega \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u \quad (5.8)$$

Stavové rovnice vyjádřené ve stavových maticích (rce.5.9).

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{C_e}{J} \\ 0 & -\frac{C_e}{L} & -\frac{R}{L} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \quad C = [0 \quad 0 \quad 1] \quad D = [0] \quad (5.9)$$

Model motoru je řízen na úhlovou polohu, výstupem je proud. Proud vynásobený konstantou motoru  $C_e$  je moment působící na motor (rce.5.10). Momentem je ovládán motor v simulaci.

$$m = C_e \cdot i \quad (5.10)$$

Model stavového popisu motoru a model získaný z identifikace jsou v bloku Načtení modelu (obr. 39) vynásobeny a dále jsou využívány pro návrh řízení.

### 5.3.1 Metoda LQR

Pro návrh řízení byl zvolen Lineární kvadratický regulátor, známý pod označením LQR. Patří do skupiny stavových zpětnovazebních regulátorů. Návrh vychází z požadavků na dynamiku (matice  $Q$ ) a také na spotřebu energie (matice  $R$ ). Při návrhu stavového zpětnovazebního regulátoru  $K$  ( $u = -Kx$ ) je využita minimalizující kvadratická hodnotící funkce (rce.5.11).[19]

$$J(u) = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \quad (5.11)$$

Vstupní parametry pro návrh regulátoru pomocí metody LQR jsou matice  $Q$ ,  $R$  a  $N$ . Matice  $Q$  a  $R$  jsou pozitivně definitní matice. Matice musí splňovat podmínku (rce.5.13).

$$R > 0 \text{ a } Q - NR^{-1}N^T \geq 0 \quad (5.12)$$

Relativní váhy odchylek stavových proměnných určuje matice  $Q$ , relativní váhy spotřeby energie stanovuje matice  $R$ . Při změně hodnot v matici  $Q$ , které odpovídají stavovým proměnným, se výrazně mění dynamiky regulace. Proto musely být odpovídající koeficienty zvoleny s velkou přesností.

Pro stavový popis složený pouze z modelu (rce.5.4) získaný z identifikace byly koeficienty voleny následovně (rce.5.14).

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad R = [2] \quad N = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \quad (5.13)$$

Tento typ řízení nezohledňuje reálné řízení motoru pomocí proudu. Proto byl do stavového popisu přidán model popisující motor (rce.5.8). Stavový popis motoru plus získaný stavový popis z identifikace. Pro tento model byly koeficienty matice  $Q$ ,  $R$  a  $N$  voleny následovně (rce.5.15).

$$Q = \begin{bmatrix} 400 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 6500 & 0 & 0 \\ 0 & 0 & 0 & 515 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = [5] \quad N = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.14)$$

## 5.4 Realizace řídicího algoritmu

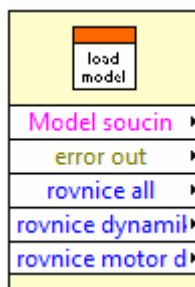
Pro realizaci řídicího algoritmu je využito nového toolkitu popsaného v předchozí kapitole. Program pro realizaci řízení modelu je sestaven ze tří základních částí, které jsou potřebné pro simulování a řízení virtuálního modelu s použitím SolidWorks. První částí je připojení a inicializace všech potřebných prvků. Základní potřebné prvky pro inicializaci jsou: Připojení k SolidWorks (Inicializace referencí), Inicializace pohybové studie, Inicializace

silového působení (moment), Synchronizace spojení (obr. 33). Dále do inicializačních prvků, potřebných pro řízení, patří Načtení modelu (obr. 39) a Návrh kontroleru (obr. 40). Dalším krokem po inicializační části je spuštění simulační smyčky. V simulační smyčce je zadáván potřebný moment a ze senzoru pozice jsou čteny rychlosti a zrychlení vozíku a motoru. Tyto údaje jsou poskytovány do LabVIEW z pohledu globálního souřadného systému v SolidWorks. Hodnoty ze senzoru jsou brány jako stavové proměnné systému, které jsou přidávány v bloku Modifikace stavu (obr. 42) do stavových rovnic. Dále je realizován regulační zásah v podobě momentu motoru, vypočítaný ze stavové proměnné proudu a konstanty motoru  $C_e$ , závislý na hodnotách matice  $Q$ ,  $R$  a  $N$ . Po stisknutí tlačítka „stop“ se zastaví simulační smyčka a je prováděna část ukončující spojení s aplikací SolidWorks. Nejdříve je zastaven výpočet pohybové studie, následně jsou ukončena silová působení a přepnuta pro použití v aplikaci SolidWorks a v poslední řadě jsou odstraněny všechny reference a je uskutečněno odpojení od aplikace SolidWorks.

Řídící algoritmus v inicializační části a hlavní simulační smyčce je sestaven z bloku realizující řízení. Podrobný popis jednotlivých bloků je popsán níže:

### Načtení modelu (Load Model)

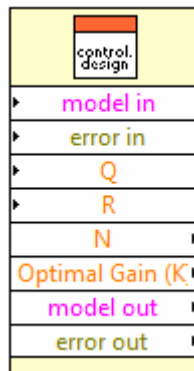
Blok načtení modelu na (obr. 39) obsahuje model stavového popisu motoru a načtení modelu, který vznikl pomocí identifikace 3D CAD modelu. Výstupem je součin těchto modelů, dále rovnice stavového popisu jednotlivých modelů a jejich součin.



Obr. 39 Načtení modelu.

### Návrh Kontroleru (Controller Design)

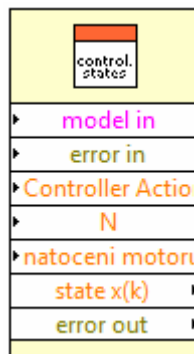
Pro řízení je potřeba navrhnout vhodný regulátor. Blok návrh kontroleru na (obr. 40) obsahuje zpětnovazební řízení LQR (Linear Quadratic Regulator), které určí optimální zesílení. Pro LQR je potřeba určit matice systému  $Q$ , matice určující dynamiku regulačního zásahu  $R$  a matice  $N$ . Výstupem bloku návrhu kontroleru je matice  $N$  a optimální zesílení.



Obr. 40 Návrh kontroleru.

### Získání stavů (Controller States)

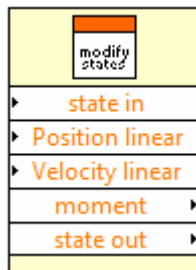
Blok zobrazený na (obr. 41), který pomocí diskretního stavového popisu získá potřebné stavy systému. Vstupem je požadované natočení motoru, matice  $N$ , zpětnovazební zásah kontroleru. Výstupem jsou požadované stavy modelu.



Obr. 41 Získání stavů.

### Modifikace stavů (Modify States)

Získané stavy z předchozího bloku je potřeba upravit. Stavy pozice a rychlost vozíku je potřeba nahradit za hodnoty čtené ze senzoru. Výstupem jsou upravené stavy a moment získaný vynásobením stavu proudu s motorovou konstantou  $C_e$ . Modifikované stavy se dále používají v bloku pro akční zásah. Blok, který provádí modifikaci stavů je na (obr. 42).



Obr. 42 Modifikace stavů.

### Akční zásah (Controller actions)

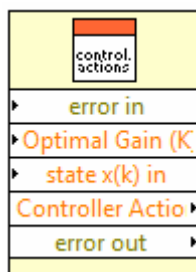
Blok na (obr. 43) slouží pro výpočet akčního zásahu. Vstupující veličiny jsou optimální zesílení, získané z bloku Návrh kontroleru, dále stavy z bloku Modifikace stavů. Akční zásah je vyjádřen v (rce.5.11). Jedná se o zpětnovazební kontroler.

$$u = -K \cdot x \quad (5.15)$$

$u$ .....akční zásah.

$K$ .....optimální zesílení.

$x$ .....stavy systému.



Obr. 43 Akční zásah.

### Podmínka (Condition)

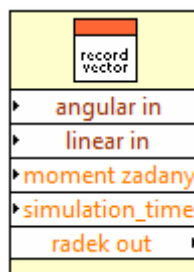
Blok pod názvem Podmínka (obr. 44) modifikuje vstupující moment plus nebo minus o danou konstantu v závislosti na požadované a měřené poloze vozíku. Operace se provádí z důvodu optimalizace hodnoty momentu vstupujícího do soustavy.



Obr. 44 Podmínka.

### Čtení (Record vector)

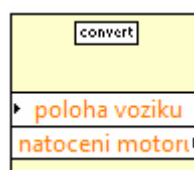
Pomocný blok (obr. 45) sloužící pro vytvoření pole všech výstupních veličin. Vstupy jsou simulační čas, hodnoty pozice, rychlosti a zrychlení motoru a vozíku a žádaný moment. Výstupní pole se dále používá pro uložení dat do textového souboru. Data z textového souboru byla využita například pro potřeby identifikace.



Obr. 45 Čtení dat ze senzorů.

### Převod polohy vozíku na natočení motoru (Convert)

Regulátor je navržen na řízení polohy motoru, proto je potřeba použít převod z polohy vozíku na natočení motoru, abychom řídili vozík na požadovanou polohu. Blok pro převod je znázorněn na (obr. 46) Výstupní natočení motoru je dále použito u bloku Získání stavů.

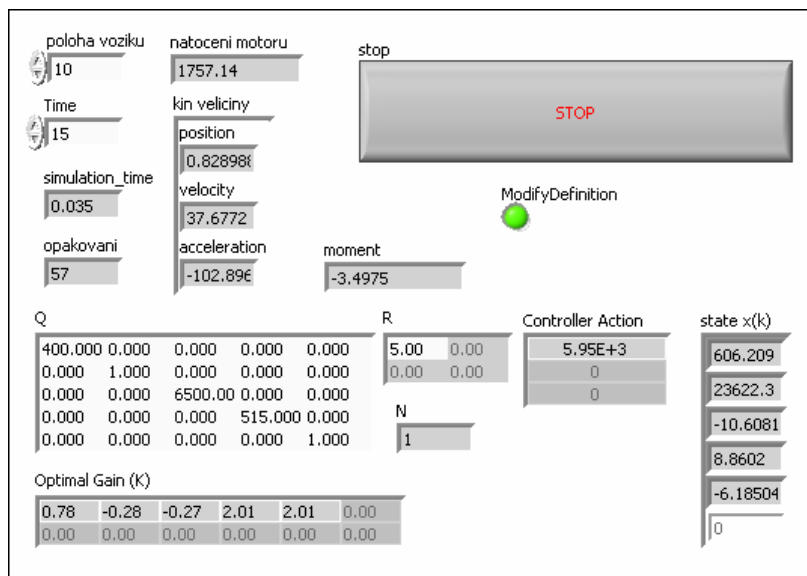


Obr. 46 Převod polohy vozíku na natočení motoru.

## 5.5 Simulace řízení na virtuálním modelu

Pro simulaci řízení virtuálního modelu byly použity bloky popsané výše. Bylo vytvořeno rozhraní, které umožňuje zadávání požadované polohy vozíku a času, potřebného pro simulaci pohybové studie a pro potřeby řízení modifikace matic  $Q$  a  $R$ . Sensory

umožňují číst kinematické veličiny vozíku. Tyto kinematické veličiny, jako jsou poloha vozíku, rychlost vozík a moment řídicí pohyb, jsou znázorněny v grafu.

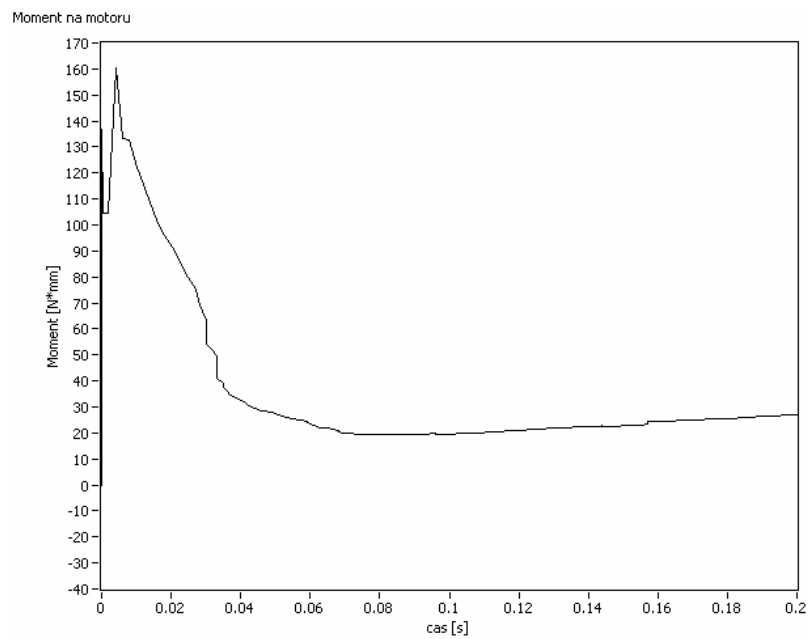


Obr. 47 Rozhraní pro řízení virtuálního modelu lineárního vozíku.

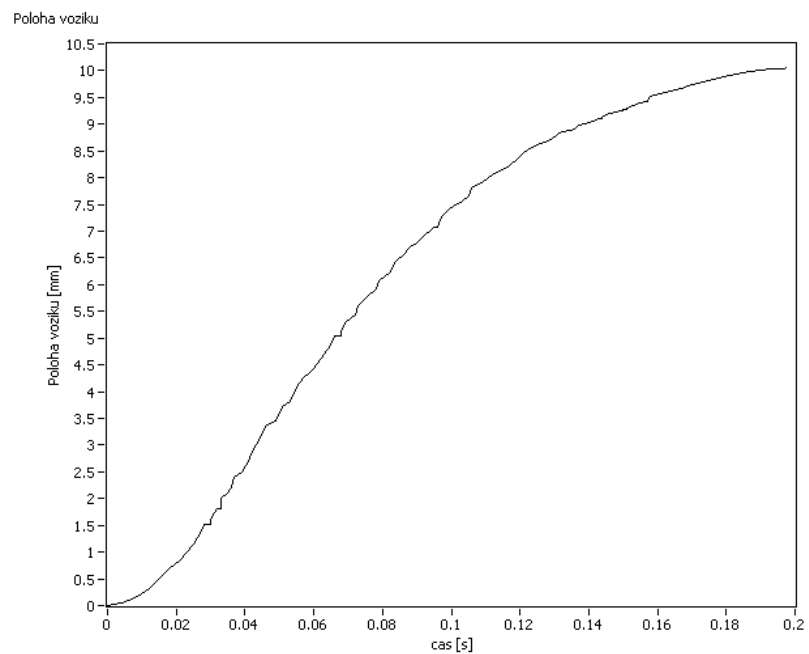
Pro spuštění programu řízení virtuálního modelu s propojením LabVIEW a SolidWorks je zapotřebí nejdříve spustit projekt v LabVIEW a následně spustit výpočet pohybové studie v SolidWorks. Po inicializaci všech komponent se přejde na fázi simulační smyčky, kde program čte data ze senzoru a provádí řízení na požadovanou hodnotu polohy vozíku. Ve smyčce je nutno ošetřit nerovnoměrné vzorkování získávání dat z pohybové studie. Vzorkování dat je dáno složitostí výpočtu pohybové studie. SolidWorks nedokáže při výpočtu pohybové studie posílat potřebná data ze senzorů. Nakonec jsou snímaná data uložena v textové podobě pro možnosti dalšího využití. Textový soubor obsahuje data v následující posloupnosti: simulační čas, poloha vozíku, rychlost vozíku, zrychlení vozíku, natočení motoru, úhlová rychlost motoru, zrychlení motoru a moment.

V následujících grafech jsou znázorněny odezvy na požadovanou hodnotu polohy vozíku rovnající se 10mm. První graf (obr. 48) znázorňuje závislost momentu na čase. Hodnota momentu je hodnota proudu vynásobená motorovou konstantou  $C_e$ . Druhý graf (obr. 49) znázorňuje závislost požadované polohy vozíku na čase. Požadovaná poloha vozíku je dosažena bez překmitu. Poslední graf (obr. 50) znázorňuje závislost rychlosti vozíku na čase. Hodnoty času jsou v jednotkách sekund.

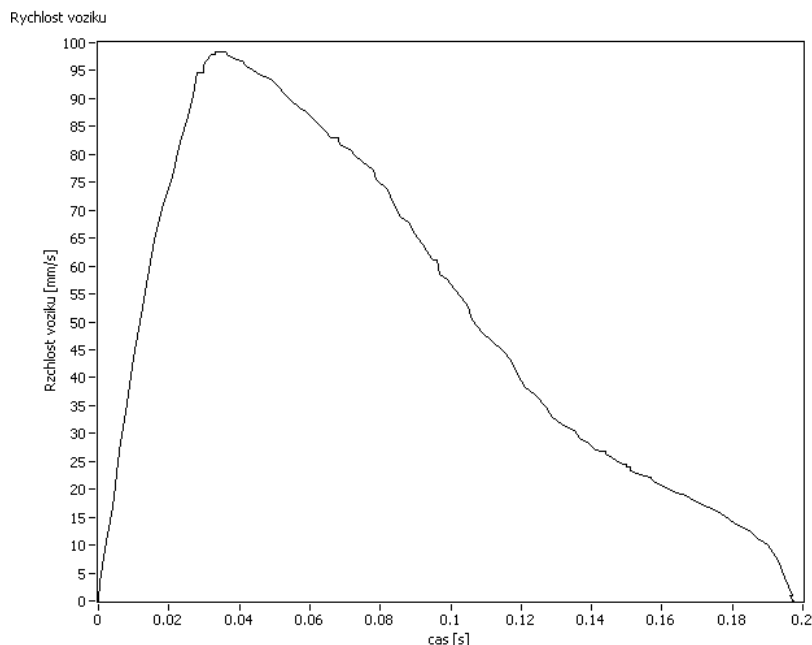




*Obr. 48 Moment na motoru.*



*Obr. 49 Poloha simulovaného voziku.*



Obr. 50 Rychlost simulovaného vozíku.

## 5.6 Realizace řídicího algoritmu na reálném modelu

### 5.6.1 Reálný model

Pro řízení reálného modelu je nezbytné mít skutečný model a veškerou potřebnou elektroniku pro řízení. Reálný model lineárního pohonu (obr. 51) byl již zkonstruován a sestaven. Reálný model je totožný s virtuálním modelem, který byl namodelován v 3D CAD prostředí SolidWorks. Základní funkcí modelu je převod rotačního pohybu z motoru na lineární pohyb za pomoci ozubené matice s kuličkovým šroubem. Prostřednictvím motoru se řídí poloha vozíku na lineárním vedení. Motor použitý na reálném modelu, který je spojen s převodovkou, pochází od firmy Maxon, Motor dále obsahuje enkodér. Použité typy jsou následující:

- DC motor Maxon RE40 (model 148877).  
 $U = 48V$  – jmenovité napětí.  
 $R = 1.16\Omega$  – odpor vinutí.  
 $L = 3.29e^{-4}H$  – indukčnost.  
 $C_e = 3.797e^{-1}$  – motorová konstanta.  
 $J = 1.34kgm^2e^{-5}$  – moment setrvačnosti.
- Planetová převodovka GP 32 C (model 166930).  
 $i = 3.7 : 1$  – převodový poměr.
- Encoder MR, type L (model 225787).  
 Počet count/ot.=1024.



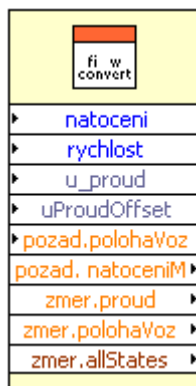
použitelné i pro řízení reálného modelu. Část bloků starající se o obsluhu SolidWorks je nahrazena za speciální bloky, které obsluhují program nahraný v FPGA.

Prvním krokem je naprogramování vnitřní program, který bude běžet na FPGA. Program se stará o propojení daných signálů. Zde se zadává, na kterém kanálu je připojen určitý signál. Program pracuje s pulzně šířkovou modulací (PWM), čtením z enkodérů, je zde také naprogramován filtr pro měření napětí a další.

Program pro řízení reálného modelu lineárního vozíku obsahuje tři základní části. První je otevření FPGA reference a počáteční nastavení potřebných parametrů. Tato část obsahuje také načtení modelu a návrh kontroleru pro potřeby řízení popsané výše. Další částí je hlavní smyčka, kde jsou čtena data z enkodéru, jako je rychlost, natočení motoru a napětí, které je přepočítáno na proud. V této smyčce se provádí řízení. Zde jsou použity stejné bloky jako pro řízení virtuálního modelu. Dále jsou přidány bloky Převod natočení a rychlosti a Modifikace duty. Tyto bloky jsou podrobně popsány níže. Akční zásah je na motor přiveden jako hodnota „duty“ (střídy). Poslední částí je nastavení hodnoty „duty“ na nulu a ukončení FPGA referencí. Hlavní část je zjednodušena o podmínku, která se stará o časově nerovnoměrné získávání dat, protože data jsou získávána prostřednictvím real-time PC (NI c-RIO).

### Převod natočení a rychlosti

Blok slouží pro převod natočení a rychlosti snímané z enkodéru na natočení a rychlosti pastorku. Dále je zde přepočet požadované polohy vozíku na požadované natočení pastorku. Vstupující napětí z enkodéru je nutno přepočítat na proud. Výstupem jsou veškeré stavy systému potřebné pro realizaci řízení. Stavy systému jsou: natočení a rychlost pastorku, proud, poloha a rychlost vozíku.



Obr. 52 Převod natočení a rychlosti.

### Modifikace duty

Blok obsahuje přepínání mezi manuálním ovládním motoru a ovládním pomocí řízení regulátoru. Vstupem pro manuální ovládní je hodnota „duty“. Pro řízení regulátorem vstupuje do bloku akční zásah, který je přepočten na hodnotu „duty“. Rozlišení enkodéru je 11bitů tzn. přibližně 2000 dílků. Z toho vyplývá, že rozsah hodnoty „duty“ je od -2000 do 2000 dílků. Výstupem je hodnota „duty“ znázorněna v grafu.

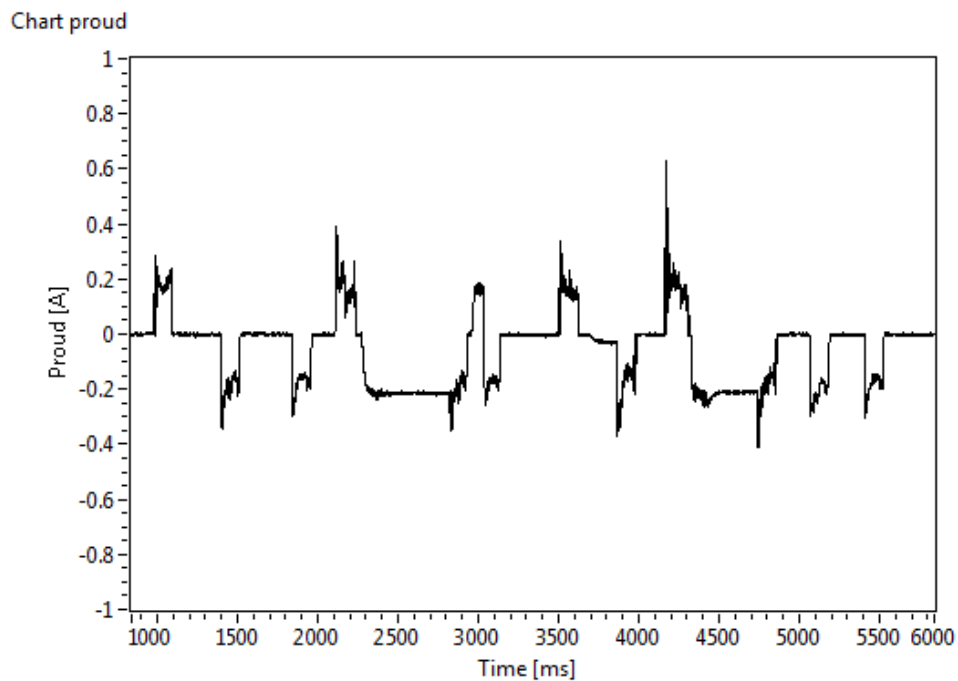


Obr. 53 Modifikace duty.

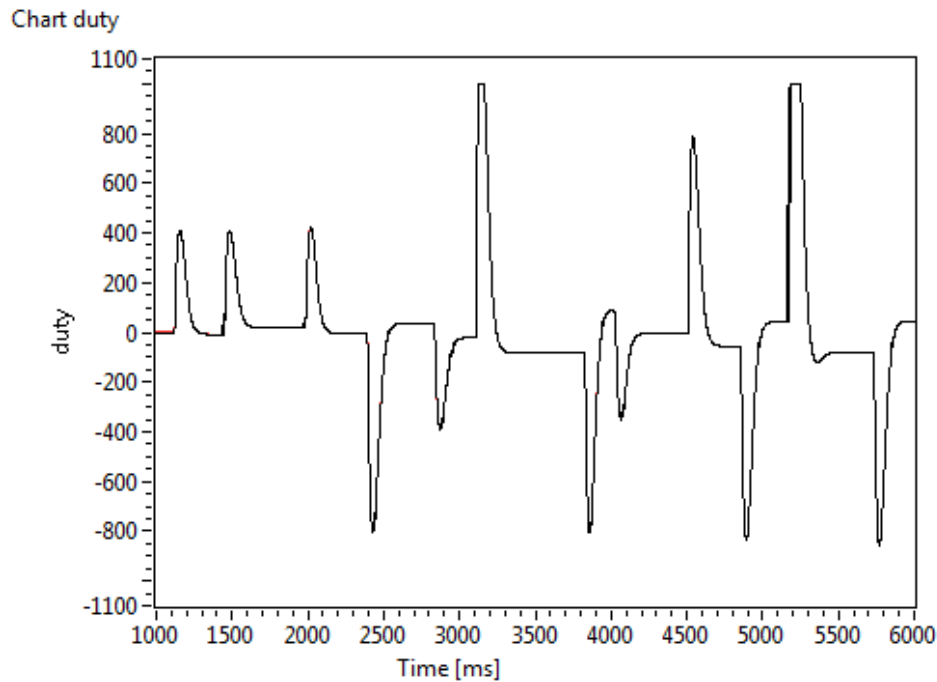
Pro řízení na reálném modelu byly použity parametry LQR regulátoru ze simulace a bylo vyzkoušeno, zda lze získanými parametry ze simulace řídit reálný model. Bylo zjištěno, že koeficienty matic  $Q$  a  $R$  (rce.5.14) je potřeba modifikovat, aby se docílilo přesné regulace. Koeficienty matic byly změněny na následující hodnoty:

$$Q = \begin{bmatrix} 700 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 6500 & 0 & 0 \\ 0 & 0 & 0 & 700 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad R = [5] \quad N = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.16)$$

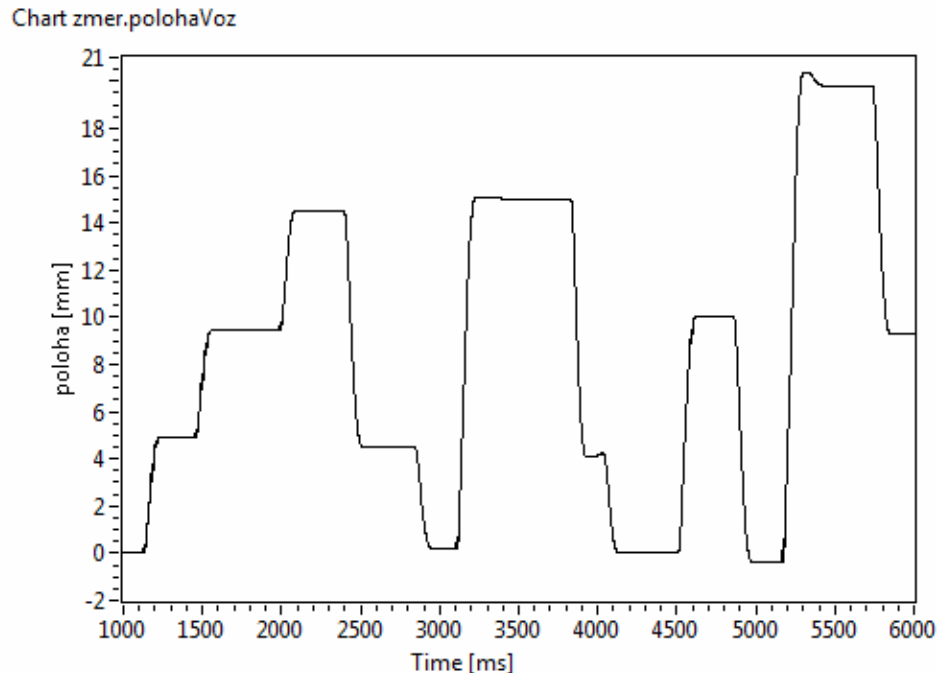
Po změně koeficientů matic jsou odezvy na požadovanou hodnotu polohy reálného vozíku znázorněny v následujících grafech. Graf na (obr. 54) označuje závislost změny proudu na čase. Další graf na (obr. 55) označuje akční zásah v podobě hodnoty „duty“ závislý na čase. Poslední graf na (obr. 56) znázorňuje požadovanou polohu reálného vozíku v závislosti na čase. Čas je uveden v jednotkách milisekund.



*Obr. 54 Změna proudu.*



*Obr. 55 Střída (duty).*



Obr. 56 Poloha reálného vozíku.

## 5.7 Zhodnocení

Od propojení LabVIEW a SolidWorks pomocí NI SoftMotion Module for SolidWorks bylo nutno ustoupit a přejít na vlastní propojení pomocí toolkitu, který naprogramoval Ing. Fridrichovský v diplomové práci. Toto propojení splňuje všechny požadavky potřebné pro návrh řízení virtuálního modelu. Následně byla provedena identifikace pomocí dat získaných použitým propojením ze simulace. Pro identifikaci se použila metoda ARX, která byla schopna simulovaný model identifikovat, tzn. vytvořit přibližný matematický popis. Pro návrh řízení byl použit LQR regulátor. Regulátor byl nastaven s využitím stavového popisu získaným z identifikace. Nastavení regulátoru bylo vyzkoušeno na virtuálním modelu pomocí použitého propojení. Dále byl regulátor využitý v simulaci vyzkoušen na reálném modelu. Výsledky využití nastavení regulátoru byly pro reálný model přijatelné s menší modifikací parametrů regulátoru.

Výsledkem je funkční řízení jak na simulovaném modelu, tak i na reálném modelu. Z toho vyplývá, že koncepce návrhu řízení s využitím propojení LabVIEW a SolidWorks umožňuje přibližné nastavení regulátoru, který je použitelný pro řízení reálného modelu.





## 6 Závěr

Diplomová práce se zabývala modelováním lineárního pohonu a návrhem řízení. Pro potřeby návrhu řízení bylo využito nástroje pro propojení grafického programovacího prostředí NI LabVIEW s 3D CAD programem SolidWorks. Model lineárního vozíku byl realizován v 3D CAD aplikaci SolidWorks, která má implementovaný nástroj pro simulaci pohybu. Pohybová studie zohledňuje realistické chování soustavy, proto je možné tuto koncepci propojení využít pro návrh řízení. Bylo vyzkoušeno chování modelu lineárního vozíku v nástroji SolidWorks v pohybové studii. Funkční spojení 3D CAD modelu (převod ozubeného kola a kuličkového šroubu) muselo být zjednodušeno a nahrazeno funkčními strojními vazbami, které odpovídají funkci daného spojení, protože řešič v SolidWorks nedokázal tak složité výpočty vypočítat. Následně bylo realizováno připojení pohybové studie v SolidWorks do prostředí LabVIEW. Pro propojení bylo využito dostupných nástrojů pod názvem NI SoftMotion Module for SolidWorks. Toto propojení bylo funkční, ale nevyhovovalo pro potřeby návrhu dynamického řízení. Model v SolidWorks mohl být stimulován jen pomocí kinematických veličin. Nebyla možnost model ovládat za pomoci momentu. Další zjištěnou nevýhodou byla nemožnost získávat ze senzorů hodnoty pozice a rychlosti vozíku pohybující se po lineárním vedení. Proto bylo nutné od dostupného propojení NI SoftMotion Module for SolidWorks odstoupit. Bylo využito toolkitu, který minulý rok vznikl při diplomové práci Ing. Fridrichovského, ještě před uvedením NI SoftMotion Module for SolidWorks na trh. Propojení pomocí toolkitu řeší všechny nedostatky propojení NI SoftMotion Module for SolidWorks. Soustava byla stimulována momentem a byly získávány potřebné odezvy simulované soustavy. Dále byla provedena identifikace systému pomocí dat získaných ze simulace. Získaný matematický popis systému byl použit pro potřeby návrhu řízení. Polohové řízení bylo realizováno za pomoci LQR regulátoru. Následně nastavený LQR regulátor pro simulaci virtuálního modelu byl také použit pro řízení reálného modelu lineárního vozíku s menší modifikací parametrů regulátoru.

Zdokonalení propojení aplikací pomocí nástroje NI SoftMotion for SolidWorks by pro možnosti návrhu dynamického řízení mělo být schopno stimulovat soustavu pomocí dynamických veličin. Dále by mělo být vyřešeno čtení dat ze senzorů.

Návrh řízení pro simulaci virtuálního modelu je dále použitelný s menšími modifikacemi pro řízení reálného modelu. Z toho plyne, že propojení nástroje určeného pro návrh řízení NI LabVIEW s nástrojem pro modelování 3D CAD systému SolidWorks je významným krokem při návrhu mechatronických soustav. Tím se urychlí doba pro potřebný vývoj mechatronických soustav, sníží se počty prototypů a klesnou náklady.



## 7 Seznam použité literatury

- [1] PELZ, Georg. *Mechatronic Systems Modelling and Simulation with HDLs*. Infineon Technologies, Munich, Germany, 2003. 236 s. ISBN 0-470-84979-7.
- [2] MAIXNER, Ladislav. *Mechatronika: učebnice*. Brno: Computer Press, 2006. 280 s. ISBN 80-251-1299-3.
- [3] PISA, Zdeněk. Simulace v systémech CAD. *MM Průmyslové spektrum* [online]. 2009, MM 2009 / 6, 19.června 2009, [cit.10.ledna 2010]. Dostupné na WWW: <http://www.mmspektrum.com/clanek/simulace-v-systemech-cad>. ISSN: 1212-2572.
- [4] Dassault Systemes SolidWorks Corp.: Understanding Motion Simulation [online]. 2008, Dostupné na WWW: [http://www.solidworks.com/sw/1867\\_ENU\\_HTML.htm](http://www.solidworks.com/sw/1867_ENU_HTML.htm)
- [5] LOMBARD, Matt *SolidWorks 2009 Bible*. 1. vyd. Indianapolis (Indiana). Wiley Publishing, Inc. 2009, ISBN: 978-0-470-25825-5
- [6] FRIDRICHOVSKÝ, Jan. *Modelování a řízení mechatronických soustav v SolidWorks a NI LabVIEW: diplomová práce* Brno: VUT v Brně, Fakulta strojního inženýrství, 2009. Vedoucí diplomové práce Ing. Pavel Houška Ph.D.
- [7] NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. Ostrava : MONTEX a.s. , 1999. 276 s. ISBN 80-7225-030-2
- [8] National Instrument: LabVIEW 2009 SP1 NI SoftMotion module Help[online]. Prosinec 2009, Dostupné na WWW: <http://digital.ni.com/manuals.nsf/websearch>
- [9] SolidWorks: SolidWorks Web Help [online]. Březen 2010, Dostupné na WWW: <http://help.solidworks.com/>
- [10] Stewart, D., *A platform with six degrees of freedom*, Proc. Inst. Mech. Eng., Vol.180, part1(15), 1965-1966, pp.371-386.
- [11] The MathWorks: About the Stewart Platform [online] 2004, poslední revize 2010[cit. 22. března 2010]. Dostupné na WWW: <http://www.mathworks.com/access/helpdesk/help/toolbox/phymod/mech/ug/f15-35077.html>
- [12] FONG, Terrence W. *Design and Trstiny of a Stewart Platform Augmented Manipulátor for Space Applications*. Massachusetts: Massachuttes Institute of Technology, June 1990, 260 s.
- [13] SINGULE, Vladislav. Motory maxon pro aplikace v humanoidní robotice a biomechanice. *T+T Technika a Trh* [online]. CCB, spol. s r.o. 22.4. 2009 . [cit. 18. března 2010]. Dostupné na WWW: [http://www.technikaatrh.cz/index.php?sec=rubrika&id\\_rubrika=45&start=3](http://www.technikaatrh.cz/index.php?sec=rubrika&id_rubrika=45&start=3) . ISSN: 1802-6176.
- [14] National Instruments: LabVIEW System Identification Toolkit User Manual[online]. Srpen 2006, Dostupné na: <http://digital.ni.com/manuals/>
- [15] ERTL, L. *Modelování a návrh řízení mechatronických soustav ve stavovém prostoru*. Brno: VUT v Brně, Fakulta strojního inženýrství, 2009. 48 s. Vedoucí bakalářské práce Ing. Pavel Houška, Ph.D.
- [16] ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M. *Automatické řízení*. 1. vyd. VUT v Brně: Akademické nakladatelství CERM s.r.o Brno, 2007, 324 s., ISBN 978-80-214-3491-2

- [17] The MathWorks: SimMechanics [online] 2004, poslední revize 4.10. 2008[cit. 22. března 2010]. Dostupné na WWW: <http://www.mathworks.co.uk/products/simmechanics/>
- [18] National Instruments: Getting started with softmotion [online]. Srpen 2006, Dostupné na WWW: <http://digital.ni.com/manuals.nsf/websearch>
- [19] SKALICKÝ, J. *Stavové řízení elektrických pohonů*. FEKT VUT v Brně, 2009
- [20] GREPL, R. *Modelování mechatronických systémů v Matlab SimMechanics* 1.vyd. nakladatelství BEN – technická literatura, Praha 2007, 152 s., ISBN 978-80-7300-226-8
- [21] The MathWorks: Simulink 7 Getting Started Guide [online] 2010, poslední revize březen 2010 [cit. 10. května 2010]. Dostupné na WWW: <http://www.mathworks.com/products/simulink/>
- [22] National Instruments: LabVIEW Control Design User Manual[online]. June 2009, Dostupné na: <http://www.ni.com/manuals/>
- [23] Maxon Motor ag: Catalog Maxon Motor [online]. 2010, [cit. 20.5.2010]. Dostupné na WWW: <http://www.maxonmotor.com/>
- [24] BALARA, M. Matematický model dynamických vlastností ložiskového reduktora Twinspin. *AUTOMA*, 2002, č. 5, s. 49-51. Dostupné na WWW: <http://www.odbornecasopisy.cz/> ISSN 1210-9592
- [25] Control Tudory Pro: Introduction M-files in Matlab.[online]. Srpen 2008, [cit. 20.5. 2010] Dostupné na WWW: <http://wikis.controltheorypro.com/>
- [26] GNU: Octane. [online].2006, [cit. 20.5. 2010] Dostupné na WWW: <http://www.gnu.org/software/octave/>