

České vysoké učení technické v Praze
Fakulta strojní

Ústav mechaniky, biomechaniky a mechatroniky
Obor: Mechatronika



**Automatizovaná tvorba
dynamických modelů prostorových
tensegritických struktur**

**Automatic model generation of
dynamic models of spatial
tensegrity structures**

DIPLOMOVÁ PRÁCE

Vypracoval: Martin Prokop
Vedoucí práce: Ing. Jan Zavřel, Ph.D.
Rok: 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Prokop** Jméno: **Martin** Osobní číslo: **483277**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Aplikované vědy ve strojním inženýrství**
Specializace: **Mechatronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Automatizovaná tvorba dynamických modelů prostorových tensegritických struktur

Název diplomové práce anglicky:

Automatic model generation of dynamic models of spatial tensegrity structures

Pokyny pro vypracování:

- 1) Seznamte se s modelováním prostorových tensegritických struktur.
- 2) Navrhněte metodiku sestavování analytického dynamického modelu tensegritické struktury.
- 3) Vytvořte softwarový nástroj pro automatické sestavení dynamického modelu na základě vstupních dat.
- 4) Ověřte vytvořený model vybrané struktury s modelem vytvořeným v prostředí MATLAB-SIMULINK-SIMSCAPE.
- 5) Zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

- [1] R. E. Skelton a M. C. Oliveira. Tensegrity Systems. Springer, Boston, 2009. ISBN 978-0-387-74241-0
- [2] Stejskal V., Valášek M.: Kinematics and dynamics of machinery. New York: Marcel Dekker, 1996. ISBN 08-247-9731-0

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Zavřel, Ph.D. odbor mechaniky a mechatroniky FS

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **18.04.2023**

Termín odevzdání diplomové práce: **14.08.2023**

Platnost zadání diplomové práce: _____

Ing. Jan Zavřel, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Michael Valášek, DrSc.
podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

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

V Praze dne

.....

Martin Prokop

Poděkování

Děkuji vedoucímu práce Ing. Jan Zavřel, Ph.D. za cenné rady a pomoc při tvorbě mé diplomové práce. Dále děkuji své rodině za obrovskou podporu při mém studiu, obzvláště mému dědovi. Děkuji i mé přítelkyni Majdě, že to se mnou zvládla až do konce.

Martin Prokop

Název práce:

Automatizovaná tvorba dynamických modelů prostorových tensegritických struktur

Autor: Martin Prokop

Studijní program: Aplikované vědy ve strojním inženýrství

Druh práce: Diplomová práce

Vedoucí práce: Ing. Jan Zavřel, Ph.D.
Odbor mechaniky a mechatroniky

Bibliografické údaje: Počet stran: 89
Počet obrázků: 28
Počet tabulek: 8
Počet příloh: 2

Abstrakt: Práce se zabývá možností automatické tvorby dynamických modelů 3D tensegritických struktur za použití fyzikálních souřadnic. Byl vytvořen software umožňující simulaci dynamiky soustav více těles. Práce je zaměřená na modelování systémů těles spojených sferickými vazbami, lany, pružinami a tlumiči, tedy prvky, kterými jsou tvořeny tensegritické struktury. Dvě takové struktury jsou v práci simulovány a porovnány s výsledky z prostředí SIMSCAPE MBS.

Klíčová slova: Tensegrity, Automatické generování modelů, Simulace 3D dynamických modelů

Title:

Automatic model generation of dynamic models of spatial tensegrity structures

Author: Martin Prokop

Abstract: The thesis deals with the possibility of automatic construction of equations of motion of 3D tensegrity systems using physical coordinates. Software has been developed to simulate the dynamics of multibody systems. The thesis focuses on modelling systems of bodies connected by spherical joints, cables, springs and dampers, i.e. the elements by which tensegrity structures are formed. Two such structures are simulated in the thesis and compared with results from the SIMSCAPE MBS software environment.

Key words: Tensegrity, Automatic model generation, 3D dynamical model simulation

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	10
Seznam tabulek	11
Cíle práce	12
Úvod	13
1 Teoretický základ	15
1.1 Krátká historie vývoje dynamických modelů	15
1.2 Newton-Eulerovy rovnice	18
1.3 SIMULINK - SIMSCAPE	19
2 Návrh metodiky pro sestavování analytického dynamického modelu tensegritické struktury	20
2.1 Matematické nástroje	20
2.1.1 Popis bodu v prostoru	20
2.1.2 Použité značení vektorů	21
2.1.3 Transformace mezi souřadnicovými systémy	21
2.1.4 Popis tělesa v prostoru	22
2.1.5 Derivace matice přechodu	23
2.1.6 Operátor vektorového součinu	24
2.1.7 Cardanovy úhly	24
2.2 Pohybové rovnice tělesa bez vazeb	25
2.2.1 Odvození hybnosti jednoho tělesa	25
2.2.2 Odvození momentu hybnosti tělesa	27
2.2.3 Newton - Eulerovy rovnice	30
2.3 Kinematické rovnice	34
2.3.1 Přepočítání vektoru rychlostí na derivaci vektoru souřadnic	34
2.3.2 Rovnice kinematiky	36
2.4 Vazbové reakční síly	36
2.5 Pohybové rovnice tělesa s vazbami	40
2.6 Pohybové rovnice pro soustavu tuhých těles	41
2.7 Porušení vazbových rovnic	43
2.8 Silové účinky od pružin, tlumičů a lan	43
2.8.1 Výpočet distribuční matice pro pružiny a tlumiče	44
2.8.2 Výpočet distribuční matice pro lano spojující více uzlů	46
2.8.3 Výpočet velikosti sil v pružinách	47
2.8.4 Výpočet velikosti sil v tlumičích	48
2.8.5 Výpočet velikosti sil v lanech	48
2.9 Odvození Jacobiho matice pro sférickou vazbu	49

3	Vytvoření softwarového nástroje pro automatické sestavení dynamického modelu tensegritické struktury	52
3.1	Vstupy softwaru	52
3.1.1	Vstupy těles	52
3.1.2	Vstupy bodů	53
3.1.3	Vstupy lan, pružin a tlumičů	53
3.1.4	Vstupy kloubů	54
3.2	Struktura softwaru	54
3.2.1	Inicializace softwaru	54
3.2.2	Vykreslení aktuální polohy	55
3.2.3	Simulace dynamiky struktury	56
3.2.4	Výsledky simulace	58
3.2.5	Animace výsledku	58
4	Ověření vytvořeného modelu vybrané tensegritické struktury s modelem vytvořeným v prostředí MATLAB-SIMULINK-SIMSCAPE	59
4.1	Dvojkyvadlo	59
4.1.1	Vstupní soubory softwaru	61
4.1.2	Model dvojkyvadla v prostředí SIMSCAPE	62
4.1.3	Porovnání modelů dvojkyvadla	62
4.2	Platforma	64
4.2.1	Vstupní soubory softwaru	66
4.2.2	Model platformy v prostředí SIMSCAPE	67
4.2.3	Porovnání modelů platformy	68
	Závěr	72
	Seznam použitých zdrojů	73
	Přílohy	76
A	Rozšířený postup zadávání nové struktury	76
A.1	Stažení softwaru	76
A.2	Spuštění softwaru	76
A.3	Zadávání vstupů nové struktury	76
A.4	Vykreslení zadané struktury	80
A.5	Spuštění simulace	83
A.6	Animace struktury	84
A.7	Úprava konstant simulace	85
A.8	Úprava metody matlabSim	85
A.9	Úprava metody stepFK	87
A.10	Výpočet inverzní úlohy	88
B	Vytvořený software	89

Seznam použitých zkratek a symbolů

A	Matice systému těles
α	Koeficient Baumgartovy stabilizace
B	Koeficient tlumení tlumiče
B	Vektor pravé strany systému těles
b	Měrný koeficient tlumení lana
β	Koeficient Baumgartovy stabilizace
β_c	Vektor pravé strany rovnic kinematiky
DAE	Algebro-diferenciální rovnice
$\mathbf{E}_{3 \times 3}$	Jednotková matice rozměru 3×3
$\boldsymbol{\eta}$	Vektor Cardanových úhlu
F	Vnější síla
\mathbf{F}_c	Vektor sil v lanech
\mathbf{F}_d	Vektor sil v tlumičích
\mathbf{F}_s	Vektor sil v pružinách
\mathbf{f}	Vektor vnějších sil
$\bar{\mathbf{c}}\mathbf{f}$	Reakční síly na souřadnicích \mathbf{s}
$\mathbf{c}\mathbf{f}$	Reakční síly na souřadnicích \mathbf{v}
\mathbf{g}	Vektor vazbových rovnic
\mathbf{g}_s	Jacobiho matice vazbových rovnic
H	Matice přepočtu úhlové rychlosti na derivaci úhlových souřadnic
i, j, k	Jednotkové vektory ve směru os x, y a z
J	Tenzor setrvačnosti tělesa
K	Tuhost pružiny
k	Měrná tuhost lana
L	Moment hmotnosti tělesa
$\boldsymbol{\lambda}_c$	Lagrangeovy multiplikátory
l_{j0}	Volná délka lana
l_j	Momentální délka lana
M	Vnější moment síly
m	Hmotnost tělesa
MKP	Metoda konečných prvků
M	Matice hmotnosti těles
n	Jednotková normála
n_B	Počet těles
n_c	Počet vazbových rovnic

ODE	Obyčejné diferenciální rovnice
Ω	Matice úhlové rychlosti
ω	Vektor úhlové rychlosti
\mathbf{P}	Hybnost tělesa
Φ_x, Φ_y, Φ_z	Cardanovi úhly
\mathbf{q}_G	Vektor gyroskopických sil
\mathbf{R}	Vektor řešení soustavy rovnic
\mathbf{r}	Polohový vektor
\mathbf{S}_{ij}	Matice přechodu mezi souřadnicovými systémy i a j
\mathbf{s}	Vektor souřadnic těles
\mathbf{T}	Matice přepočtu vektoru rychlosti na vektor souřadnic
\mathbf{T}_c	Matice distribuce sil lan
\mathbf{T}_d	Matice distribuce sil tlumičů
\mathbf{T}_s	Matice distribuce sil pružin
\mathbf{v}	Vektor rychlostí
W	Vykonaná mechanický práce
\mathbf{x}	Stavový vektor systému

Seznam obrázků

Obr. 1	Ukázka tensegritické struktury	14
Obr. 1.1	Popis polohy tyče DAE formulace	16
Obr. 2.1	Popis polohy bodu	21
Obr. 2.2	Ukázka bázových vektorů dvou souřadnicových systémů	22
Obr. 2.3	Vztah mezi dvěma souřadnicovými systémy	23
Obr. 2.4	Vektorový diagram pro odvození hybnosti a momentu hybnosti	26
Obr. 2.5	Silový účinek na těleso	32
Obr. 2.6	Pružina mezi dvěma tělesy	44
Obr. 2.7	Lano mezi třemi tělesy	46
Obr. 2.8	Spojení dvou těles sférickou vazbou	50
Obr. 3.1	Ukázka práce se softwarem	55
Obr. 3.2	Diagram aktivit inicializace objektu TenseMech	55
Obr. 3.3	Diagram aktivit vykreslování aktuální polohy struktury	55
Obr. 3.4	Diagram aktivit metody <code>matlabsim(timeStop)</code>	56
Obr. 3.5	Diagram aktivit metody <code>stepFK(t, x)</code>	56
Obr. 3.6	Příkaz pro vykreslení souřadnic x, y, z prvního tělesa	58
Obr. 4.1	Náčrt řešeného modelu dvojkyvadla s lanem	60
Obr. 4.2	Model v prostředí SIMSCAPE MBS pro dvojkyvadlo	63
Obr. 4.3	Porovnání výsledků simulace dvojkyvadla	63
Obr. 4.4	Průběh absolutní chyby simulace dvojkyvadla	64
Obr. 4.5	Schéma 1 platformy	65
Obr. 4.6	Schéma 2 platformy	65
Obr. 4.7	Model v prostředí SIMSCAPE MBS pro platformu	67
Obr. 4.8	Simulace platformy při síle $40 N$	68
Obr. 4.9	Chyba průběhu souřadnic simulace platformy při síle $40 N$	69
Obr. 4.10	Simulace platformy při síle $5 N$	70
Obr. 4.11	Chyba průběhu souřadnic simulace platformy při síle $5 N$	70
Obr. 4.12	Průběh sil při simulaci $5 N$ platformy	71

Seznam tabulek

Obr. 4.1	Dvojkyvadlo - BODY.CSV	61
Obr. 4.2	Dvojkyvadlo - POINT.CSV	61
Obr. 4.3	Dvojkyvadlo - JOINT.CSV	62
Obr. 4.4	Dvojkyvadlo - CABLE.CSV	62
Obr. 4.5	Platforma - BODY.CSV	66
Obr. 4.6	Platforma - POINT.CSV	66
Obr. 4.7	Platforma - JOINT.CSV	66
Obr. 4.8	Platforma - CABLE.CSV	67

Cíle práce

V této části jsou uvedeny jednotlivé cíle práce.

- Prvním cílem práce je seznámení se s principy modelování prostorových tensegritických struktur spolu s navržením metodiky sestavování analytického modelu tensegritické struktury.
- Druhým cílem je navržení struktury softwaru umožňující aplikovat principy navržené v první části. Navržený software je následně třeba vytvořit.
- Posledním cílem je ověřit výsledky dynamické simulace vybraných struktur s modelem vytvořeným v prostředí MATLAB-SIMULINK-SIMSCAPE. Dosažené výsledky je třeba zhodnotit

Úvod

Tensegritické struktury nebo-li tensegrity jsou sítí osově namáhaných prvků. Pojem tensegrity vznikl zkrácením anglických slov *tension* a *integrity*. Autorem této zkratky je Buckminster Fuller [1]. Celá struktura je tvořena dvěma typy prvků, a to prvky namáhanými na tlak (dále označované jako *tyče* kvůli obvyklé realizaci v podobě tyčí) a prvky namáhanými na tah (dále označované jako *lana*). Tyto komponenty dohromady vytvářejí stabilní mechanický systém [2].

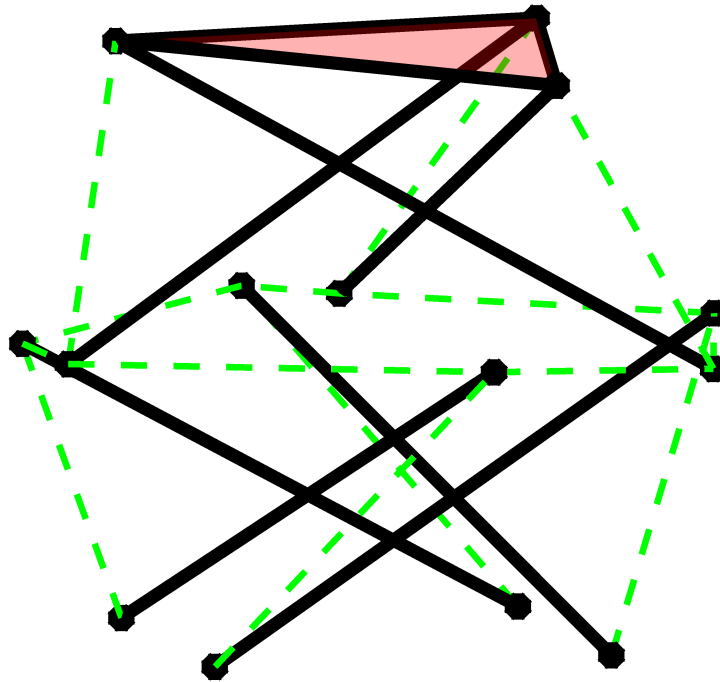
Tensegrity řadíme do *tříd* (ang. *tensegrity class*). O tensegritě říkáme, že je první třídy, pokud se žádná z tyčí nedotýká jiné. V opačném případě mluvíme o tensegritách třídy k , přičemž k je maximální počet tyčí dotýkajících se navzájem v jednom bodě prostoru. Bodům v prostoru, ve kterých jsou spojeny tyče a lana, říkáme *uzly* (ang. *nodes*). Je důležité poznamenat, že uzlem může být místo, ve kterém jsou spojeny jak tyče s lany tak tyče s tyčemi nebo lana s lany. Spojení tyče s tyčí se modeluje pomocí sférické vazby bez pasivních odporů. Díky tomu, že všechna lana prochází středy uzlů, jsou použity sférické vazby a pasivní účinky jsou zanedbány, jsou všechny prvky struktury namáhány pouze osovými silami, tedy čistým tahem a tlakem. V případech čistě osově namáhaných elementů dokážeme nejlépe předpovědět chování materiálů, a díky tomu prvky správně dimenzovat. Při vytváření reálných modelů tensegrit nedokážeme vyrobit vazbu bez pasivních účinků nebo vytvořit geometricky přesný model. Ovšem všechny vniklé mimoosové účinky od těchto nepřesností (zejména namáhání momentem síly) budou natolik malé, že je lze v reálných aplikacích zanedbat [3].

Velmi zajímavou vlastností tensegrit je jejich schopnost měnit tuhost struktury při zachování aktuální konfigurace nebo změnit konfiguraci při zachování tuhosti. Díky této vlastnosti jsou tensegrity vhodné pro aplikace v oblasti jemné robotiky [4] [5]. Oba typy změn lze provést buď překonfigurováním vnitřní části tensegrity, anebo vhodným přidáním prestressu [3, s. 23]. Tensegrity se tedy řadí do předepnutelných struktur (ang. *prestressed structure*). To jsou takové struktury, které umožňují existenci vnitřních sil i při absenci sil vnějších [6, s. 6].

Spolu s vývojem teorie tensegritických struktur je potřeba vyvíjet i nástroje pro dynamické simulace tensegrit. V posledních letech došlo k vývoji softwarových nástrojů pro automatické sestavování dynamických modelů. Jednou z významných výzkumných skupin zabývajících se problematikou automatického sestavování je sku-

pina *MOTES* (Modeling of tensegrity structures), která poskytla svůj software pro automatické sestavování simulačního modelu tensegrit psaný v prostředí MATLAB, využívající neustále se rozvíjející přístup modelování tensegrit od R. Skeltona [7]. Jiný přístup k modelování tensegrit zvolili M. Prokop [8] nebo M. Hrabačka [9], kteří pro modelování zvolili prostředí MATLAB-SIMULINK s knihovnou SIMSCAPE MBS. Takto vygenerované modely jsou díky rychlému výpočetnímu času vhodným nástrojem pro simulaci přímé dynamiky systému. Nevýhoda prostředí SIMSCAPE spočívá v tom, že uživatel nemá kontrolu nad rovnicemi popisující problém. To se ukázalo být jako nevýhoda v případech, kdy uživatel potřebuje tensegritu řídit.

Ukázku tensegritické struktury je možné vidět na obr. 1.



Obr. 1: Ukázka tensegritické struktury

Kapitola 1

Teoretický základ

U tensegritických struktur se zkoumají jejich statické a dynamické vlastnosti. Mezi statické vlastnosti se řadí úloha form-finding a vyšetřování stability struktury. U dynamických modelů je situace komplikovanější. V posledních letech stále probíhá vývoj teorií pro popis dynamického chování tensegrit. V této kapitole bude popsán vývoj dynamických modelů tensegrit. V závěru kapitoly bude vysvětlen důvod použití standardního popisu tensegrity jako multibody systému při použití fyzikálních souřadnic. Krátce bude diskutován důvod, proč nebylo zvoleno prostředí SIMSCAPE MBS.

1.1 Krátká historie vývoje dynamických modelů

Poprvé se dynamikou tensegrit zabývali v roce 1986 R. Motro, S. Najari a P. Jouanna. Autoři v článku popsali několik pojmů ze statické analýzy, jako je self-stres nebo form-finding a představili pohybové rovnice popisující dynamiku tensegrit. Prezentovali zde i výsledky z experimentální studie lineární dynamiky tensegrity složené ze tří tyčí a devíti lan známé jako SIMPLEX. Výsledky ze statické i dynamické simulace byly ve shodě s měřením reálné tensegrity. Dynamiku tensegrity popsali ve formě pohybových rovnic za použití uzlových souřadnic (přirozené souřadnice), stejně jako kdyby přistupovali k dynamické úloze modelované metodou konečných prvků (MKP) [10][11][12].

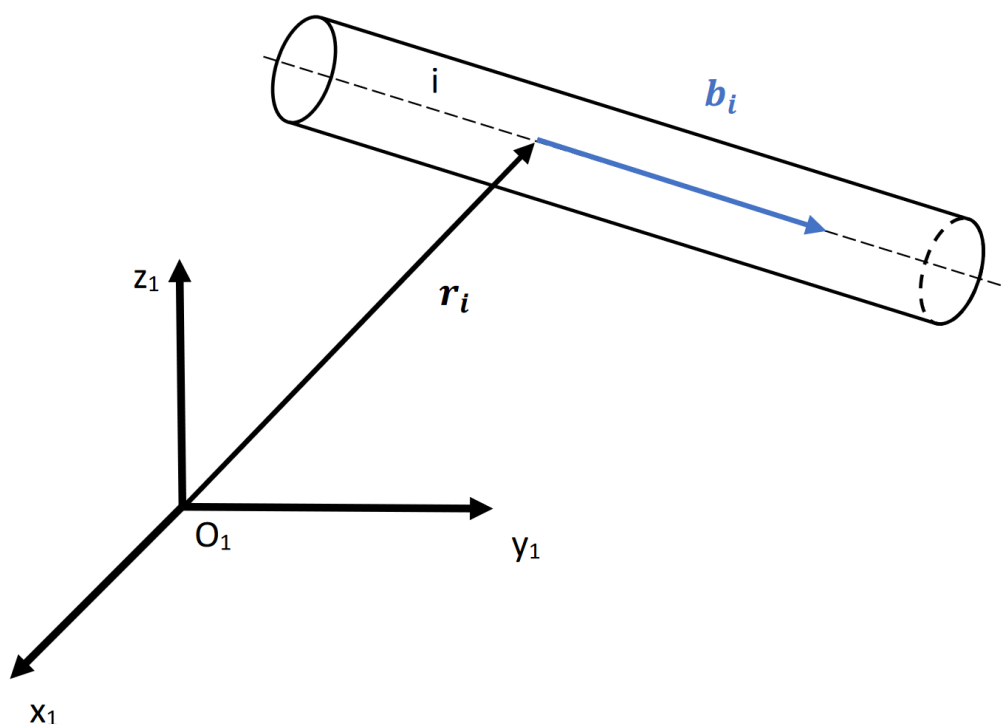
V roce 1992 Furuya použil MKP program SAP-IV pro modelování tensegrit a zjistil, že vlastní frekvence rostou se zvyšujícím se předepnutím. Tento výsledek je ve shodě s analytickým řešením vlnové rovnice lana s nulovou ohybovou tuhostí napnutého mezi dvě kloubové podpěry [12][13].

C. Sultan v roce 1999 přinesl možnost modelování tensegrit za použití nezávislých souřadnic, které vedou na soustavu nelineárních obyčejných diferenciálních rovnic (ODE ang. *ordinary differential equation*). Jeho odvozené rovnice byly použity při aplikování nelineárních řídicích zákonů pro vesmírné teleskopy [14]. O tři roky později (2002) pak C. Sultan publikoval linearizované pohybové rovnice vhodné

pro řízení lineárními zákony řízení [15].

Přestože využití ODE rovnic umožňovalo nalézt způsoby řízení tensegrit, stále obsahovalo spoustu výpočetně náročných funkcí (transcendentálních). Tento problém eliminoval v roce 2005 R. Skelton, který představil jeden z nejjednodušších tvarů dynamiky pro tensegrity třídy 1. Ve své práci představil model využívající nadbytečné souřadnice, přičemž z modelu odpadla potřeba počítat časově náročné funkce, jako je sinus, cosinus a exponenciální funkce. Za využití nadbytečných souřadnic je ovšem potřeba zaplatit cenou ve formě ztráty pohybových rovnic v ODE podobě a na místo toho pohybové rovnice formulovat ve tvaru DAE (differential-algebraic equations). Další nevýhodou použití nadbytečných souřadnic je větší dimenze řešené úlohy [16] [17].

Pro tento popis dynamiky byl použit jeden bod na tyči a jednotkový vektor orientace tyče obr. 1.1. Vzhledem k uvažované rotační symetrii tyče a zanedbané rotační setrvačnosti okolo osy symetrie, má každá z tyčí pouze pět reálných stupňů volnosti. Při použití tří souřadnic pro popis bodu na tyči (vektor \mathbf{r}_i) a tří souřadnic pro popis natočení tyče (vektor \mathbf{b}_i) je patrné, že je jedna souřadnice pro každou tyč přebývá [17].



Obr. 1.1: Popis polohy tyče DAE formulace

Složením těchto dvou vektorů vznikne vektor $\vec{q}_i \in \mathbb{R}^6$ popisující aktuální stav tyče i .

$$\mathbf{q}_i = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{b}_i \end{bmatrix} \in \mathbb{R}^6 \quad \mathbf{r}_i = \begin{bmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{bmatrix} \in \mathbb{R}^3 \quad \mathbf{b}_i = \begin{bmatrix} b_{xi} \\ b_{yi} \\ b_{zi} \end{bmatrix} \in \mathbb{R}^3 \quad (1.1)$$

Tento model byl vhodný pouze pro tensegrity první třídy. S modelováním tyčí uchycených pevně k rámu nebyl problém, ale žádné tyče se navzájem nesměly dotýkat. Při odvozování rovnic dynamiky bylo využito mnoho zjednodušujících předpokladů. Mezi ně patří požadavky na nehmotnost lan, nulovou ohybovou tuhost lan, nulovou setrvačnost tyčí okolo své vlastní osy a na model tensegrity třídy jedna. Dalším požadavkem bylo, aby lano vždy spojovalo pouze dvě tyče mezi sebou (nebylo možné vytvořit uzel spojující lana). A. S. Wroldsen pak tyto dynamické rovnice použil ve své disertační práci pro řízení jednoduchých tensegrit [16].

Použití nadbytečných souřadnic přineslo velkou výhodu v podobě eliminace singularit. Při použití Eulerových nebo Cardanových úhlů má popis natočení tělesa vždy singularity. Tyto singularity souvisí s nejednoznačným přiřazením Eulerových nebo Cardanových úhlů k matici transformace. Singularity se při použití Skeltonova popisu neobjeví [3] [16].

V roce 2009 Skelton v publikaci *Efficient dynamic models of tensegrity systems* transformoval dynamiku polohy a natočení tyčí (rovnice (1.1) a obr. 1.1) na dynamiku uzlů [18]. Konstantní matice hmotnosti v kombinaci s možností počítání dynamiky bez nutnosti použití transcendentních funkcí umožnila rychlé a efektivní výpočty struktur třídy 1. Od té doby jsou všechny výsledné dynamické modely psané v přirozených souřadnicích. Při odvození rovnic dynamiky se však stále vychází z popisu tyčí za pomoci bodu a jednotkového vektoru orientace [12] [18].

J. Cheong, R. Skelton a Y. Cho v roce 2015 představili způsob, jak modelovat tensegrity vyšších tříd (třídy k). Doplňovali rovnice dynamiky uzlů o Lagrangeovy multiplikátory a odvodili algebraickou rovnici, ze které je možné multiplikátory dopočítat. Dále představili způsob redukce počtu souřadnic pomocí SVD rozkladu a odvodili algoritmus pro korekci délek tyčí. Rovnice dynamiky nyní bylo možné aplikovat i na tensegrity vyšších tříd. Hlavní problém pro dynamický model v této fázi byl, že rovnice pro výpočet Lagrangeových multiplikátorů se musela řešit numericky. Neexistoval vztah pro jejich výpočet v uzavřeném tvaru [12].

R. Goyal a R. Skelton v roce 2019 ukázali, jak do modelu přidat lanům hmotnost. To bylo žádané pro statiku i dynamiku struktur využívajících masivní lana, jako jsou mosty, kde hmotnost lan není zanedbatelná. Dále našli algoritmizovatelný způsob modelování uzlů spojující pouze lana v uzlu. Díky tomu je možné lana rozdělit na více částí a věrohodněji simulovat jejich dynamiku. Podařilo se jim zefektivnit jejich algoritmus pro korekce délek tyčí a našli řešení rovnice pro výpočet Lagrangeových multiplikátorů v uzavřeném tvaru. Řešení rovnice pro multiplikátory

se ukázalo být lineární. Jejich výpočet tedy do úlohy nevnáší nelinearity [19].

V roce 2020 R. Goyal, M. Majji a R. Skelton publikovali článek zabývající se vlivem gyroskopických sil od rotujících kotoučů na tyčích. Přidáním kotoučů ovšem porušili původní předpoklady o zanedbatelném momentu setrvačnosti tyčí a bylo třeba gyroskopické síly do modelu přidat. Přidání těchto sil umožňuje řízení tvaru tensegrity i v rovinách bez řízených lan, zlepšuje dosažitelnost konstrukce a zvyšuje redundanci řízení [20]. Ve stejném roce poté model otestovali na reálné tensegritě a naměřili výsledky ve shodě s modelem [21].

Skupina MOTES (Modeling of Tensegrity Structures) výsledné dynamické rovnice implementovala do svého volně dostupného softwaru publikovaného na síti GitHub. Autory softwaru jsou R. Goyal, M. Chen, M. Majji a R. Skelton [7].

1.2 Newton-Eulerovy rovnice

Diferenciální rovnice odvozené Skeltonem a kol. dávají jasný postup, jak dynamiku tensegrit algoritmizovat tak, aby bylo možné popsat tensegritické struktury libovolných rozměrů. Na druhou stranu jsou tyto rovnice přímo šité na míru tensegritám a je přinejmenším obtížné do nich přidat jiný prvek, než pro který byly rovnice odvozeny. Takovým prvkem může být například end-efektorová platforma připojená k tensegritě. Tu sice lze modelovat jako tři tyče navzájem spojené sférickou vazbou, ale model se začíná dostávat na hranici svých možností. Pokud by bylo třeba modelovat end-efektorovou plošinu s jinými hmotnostními charakteristikami, zejména s jinými momenty setrvačnosti, než které vytvoří spojení tří tyčí, objevil by se problém, jak to udělat. Z toho důvodu bylo rozhodnuto jít cestou modelování tensegrit jako multi-body systému popsáno fyzikálními souřadnicemi.

Takto popsaný dynamický model obsahuje spoustu transcendentálních funkcí, kterým se Skeltonův model vyhnul. Při dnešních výpočetních výkonech počítačů není použití časově náročnějších funkcí takový problém, jako před osmnácti lety, kdy se Skeltonovy modely začaly rozvíjet.

Použití fyzikálních souřadnic je ovšem velkou výhodou čitelnosti modelu. Dynamické rovnice se odvozují za pomoci Newton-Eulerových rovnic, Lagrangeových rovnic nebo Hamiltonových rovnic.

Cílem této práce je vytvoření softwaru pro modelování tensegritických struktur s možností spojení k jiným objektům, jako je end-efektorová platforma. Toho by se s připraveným softwarovým nástrojem od skupiny MOTES dosahovalo obtížně. Výsledný software musí umožňovat modelovat tensegritické struktury i jejich spojení s jinými tělesy.

1.3 SIMULINK - SIMSCAPE

Je důležité také zmínit důvod, proč pro automatické sestavování modelu nebylo zvoleno prostředí SIMSCAPE MBS. To umožňuje výpočet přímé dynamické úlohy, ale s inverzní dynamickou úlohou začíná být problém. SIMSCAPE nepodporuje např. výpočet inverzní dynamiky pro soustavy s redundantními pohony. Vzhledem k tomu, že vytvářený software má být v budoucnu použit i pro inverzní úlohy s redundantními pohony, je využití prostředí SIMSCAPE nemožné.

Prostředí SIMSCAPE bude použito v této práci pro verifikaci dynamického modelu sestaveného vytvářeným softwarovým nástrojem na úlohách přímé dynamiky.

Kapitola 2

Návrh metodiky pro sestavování analytického dynamického modelu tensegritické struktury

V této kapitole budou odvozeny pohybové rovnice popisující dynamiku systému více těles pro následnou implementaci ve vytvářeném softwaru.

2.1 Matematické nástroje

Před začátkem odvození rovnic dynamiky je potřeba představit použitá značení.

2.1.1 Popis bodu v prostoru

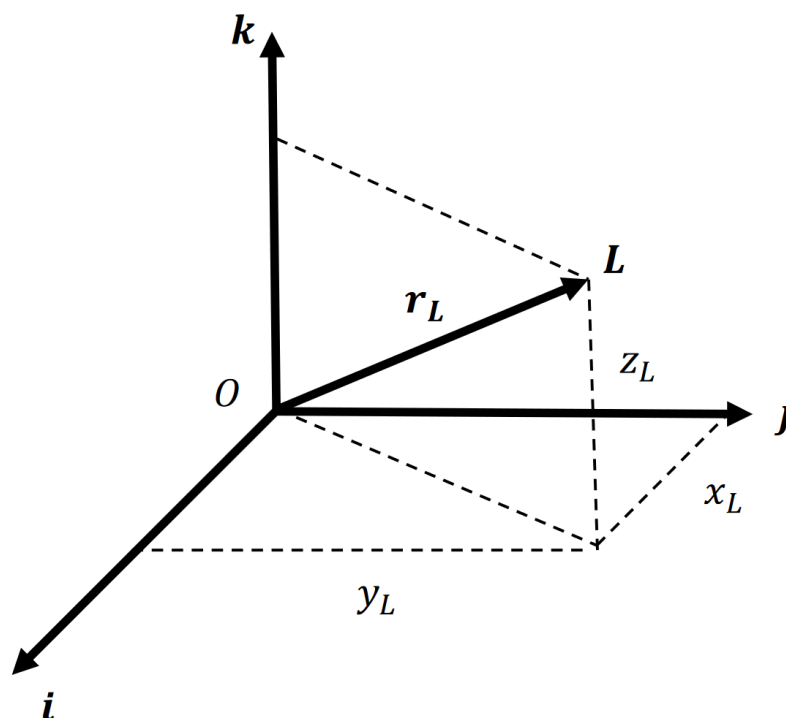
Pro popis polohy bodu L v prostoru je použit polohový vektor \mathbf{r}_L (viz obr. 2.1), který vyjadřuje polohu bodu vzhledem k souřadnicovému systému. Ten je tvořen bázovými jednotkovými vektory \mathbf{i} , \mathbf{j} a \mathbf{k} . Vektor \mathbf{i} směřuje v kladném směru osy x , vektor \mathbf{j} směřuje v kladném směru osy y a vektor \mathbf{k} směřuje v kladném směru osy z . Polohový vektor \mathbf{r}_L je vyjádřen jako lineární kombinace těchto bázových vektorů s koeficienty x_L , y_L a z_L :

$$\mathbf{r}_L = x_L \mathbf{i} + y_L \mathbf{j} + z_L \mathbf{k} \quad (2.1)$$

Vektory \mathbf{i} , \mathbf{j} a \mathbf{k} tvoří ortonormální souřadnicový systém. Platí následující vztahy:

$$\mathbf{i}^T \mathbf{i} = 1, \quad \mathbf{j}^T \mathbf{j} = 1, \quad \mathbf{k}^T \mathbf{k} = 1 \quad (2.2)$$

$$\mathbf{i}^T \mathbf{j} = 0, \quad \mathbf{j}^T \mathbf{k} = 0, \quad \mathbf{k}^T \mathbf{i} = 0 \quad (2.3)$$



Obr. 2.1: Popis polohy bodu

Vzhledem k aplikaci následujících rovnic na počítači, budou použity tzv. algebraické vektory. To je realizace vektorů za pomoci sloupcových matic [22]. Z tohoto důvodu je skalární součin dvou vektorů zapsán jako násobení transponovaného prvního vektoru s druhým. Algebraické vektory se budou značit **tučným** symbolem.

2.1.2 Použité značení vektorů

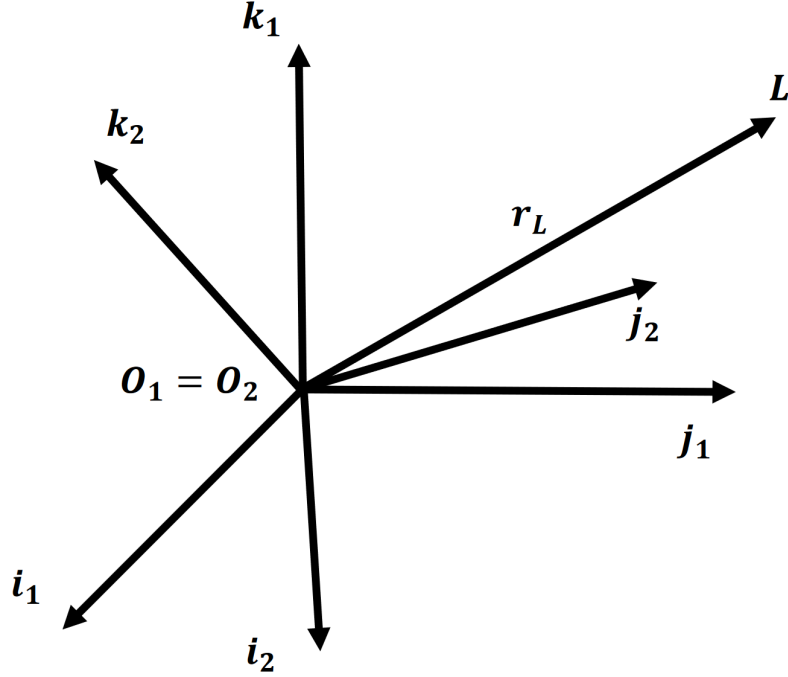
Bude použito následující značení vektorů:

$${}^i\mathbf{r}_{LT} \quad (2.4)$$

Horním levým indexem (na ukázce 2.4 značeno písmenem i) je značen souřadnicový systém, ve kterém jsou vyjádřeny složky vektoru \mathbf{r} . Dolním pravým indexem jsou značeny body, které vektor spojuje. První písmeno označuje počátek vektoru a druhé písmeno konec vektoru. Pokud je v pravém dolním indexu pouze jedno písmeno, jedná se o vektor spojující počátek souřadnicového systému i s příslušným bodem.

2.1.3 Transformace mezi souřadnicovými systémy

Pro popis polohy bodu je možné využít více různých souřadnicových systémů (obr. 2.2). Transformaci vektoru \mathbf{r}_L z jednoho souřadnicového systému do druhého se provádí za použití *matice transformace* označované také jako *matice přechodu* nebo *matice směrových cosinů* \mathbf{S}_{12} .



Obr. 2.2: Ukázka bázových vektorů dvou souřadnicových systémů

$${}^1\mathbf{r}_L = \mathbf{S}_{12} {}^2\mathbf{r}_L \quad (2.5)$$

V rovnici (2.5) matice \mathbf{S}_{12} transformuje složky vektoru \mathbf{r}_L ze souřadnicového systému označeného číslem 2 do systému s číslem 1.

Důležitou vlastností matice přechodu je její ortonormalita. Její inverze je rovna transpozici matice:

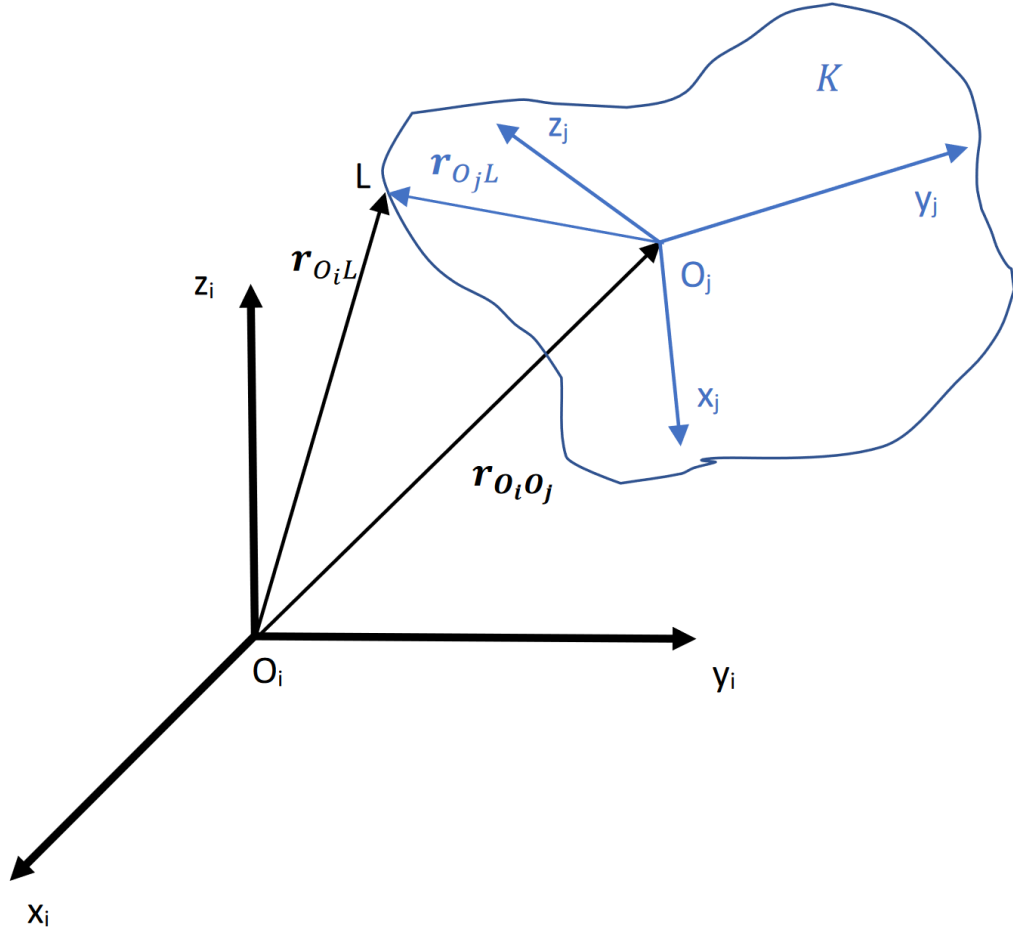
$$\mathbf{S}_{12}^{-1} = \mathbf{S}_{12}^T \quad (2.6)$$

Zároveň platí:

$$\mathbf{S}_{12}\mathbf{S}_{12}^T = \mathbf{S}_{12}\mathbf{S}_{12}^{-1} = \mathbf{E}_{3 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

2.1.4 Popis tělesa v prostoru

Pod pojmem popis tělesa v prostoru je chápán způsob, jak popsat polohu a natočení tělesa (dohromady konfiguraci tělesa) vůči jinému referenčnímu souřadnicovému systému (nejčastěji vůči pevně umístěnému rámu). Za tímto účelem jsou zaváděny lokální souřadnicové systémy pevně spojené s tělesem. K úplnému popisu konfigurace tělesa pak stačí znát vztah mezi příslušnými dvěma souřadnicovými systémy. Pro popis tělesa v prostoru je třeba znát rádiusvektor spojující referenční souřadnicový systém a lokální souřadnicový systém. Vzájemnou rotaci souřadnicových systémů pak popisuje matice transformace \mathbf{S}_{ij} popsaná v sekci 2.1.3.



Obr. 2.3: Vztah mezi dvěma souřadnicovými systémy

Jako příklad je uveden obr. 2.3. Na něm je vidět modré těleso značené písmenem K , jehož konfiguraci je za cíl popsat vůči černému referenčnímu souřadnicovému systému i . K tělesu je pevně připojen souřadnicový systém j . Poloha tělesa je popsána pomocí vektoru $\mathbf{r}_{O_i O_j}$. Natočení pak popisuje matice \mathbf{S}_{ij} , která není na obr. 2.3 zobrazena. Polohu libovolného bodu L pevně spojeného s tělesem K v prostoru lze popsat z pohledu referenčního souřadnicového systému za pomoci rádiusvektoru $\mathbf{r}_{O_i L}$. Při znalosti konfigurace tělesa je možné rádiusvektor $\mathbf{r}_{O_i L}$ vyjádřit za pomoci vektoru polohy $\mathbf{r}_{O_i O_j}$, matice přechodu \mathbf{S}_{ij} a lokálního vektoru $\mathbf{r}_{O_j L}$ jako:

$${}^i \mathbf{r}_{O_i L} = {}^i \mathbf{r}_{O_i O_j} + \mathbf{S}_{ij} {}^j \mathbf{r}_{O_j L} \quad (2.8)$$

2.1.5 Derivace matice přechodu

Důležitou vlastností matice směrových cosinů \mathbf{S}_{ij} je možnost vyjádření její časové derivace za pomoci násobení maticí úhlových rychlostí. Platí vztah:

$$\dot{\mathbf{S}}_{ij} = \mathbf{S}_{ij} {}^j \boldsymbol{\Omega}_{ij} \quad (2.9)$$

Kde matice úhlové rychlosti $\boldsymbol{\Omega}_{ij}$ je antisymetrická matice popisující rychlost

otáčení souřadnicového systému j vůči souřadnicovému systému i [22, s. 135].

$${}^j\boldsymbol{\Omega}_{ij} = \begin{bmatrix} 0 & -{}^j\boldsymbol{\omega}_{ijz} & {}^j\boldsymbol{\omega}_{ijy} \\ {}^j\boldsymbol{\omega}_{ijz} & 0 & -{}^j\boldsymbol{\omega}_{ijx} \\ -{}^j\boldsymbol{\omega}_{ijy} & {}^j\boldsymbol{\omega}_{ijx} & 0 \end{bmatrix} \quad (2.10)$$

Je důležité zmínit, že složky matice úhlové rychlosti jsou vyjádřeny v souřadnicovém systému j .

Matice ${}^j\boldsymbol{\Omega}_{ij}$ v sobě obsahuje prvky vektoru úhlové rychlosti ${}^j\boldsymbol{\omega}_{ij}$:

$${}^j\boldsymbol{\omega}_{ij} = \begin{bmatrix} {}^j\boldsymbol{\omega}_{ijx} \\ {}^j\boldsymbol{\omega}_{ijy} \\ {}^j\boldsymbol{\omega}_{ijz} \end{bmatrix} \quad (2.11)$$

2.1.6 Operátor vektorového součinu

V práci bude využito operátoru vektorového součinu. Ten je vhodný zavést při integraci výpočetních algoritmů, protože převádí problém vektorového součinu na problém násobení matice s vektorem.

$${}^i\mathbf{r}_{O_iT} = \begin{bmatrix} {}^ix_{O_iT} \\ {}^iy_{O_iT} \\ {}^iz_{O_iT} \end{bmatrix}, \quad {}^i\tilde{\mathbf{r}} = \begin{bmatrix} 0 & -{}^iz_{O_iT} & {}^iy_{O_iT} \\ {}^iz_{O_iT} & 0 & -{}^ix_{O_iT} \\ -{}^iy_{O_iT} & {}^ix_{O_iT} & 0 \end{bmatrix} \quad (2.12)$$

Všechny vektory v jejich antisymetrické formě budou značeny pomocí operátoru tilde (\sim). Např. antisymetrická matice vektoru \mathbf{r} bude značena jako $\tilde{\mathbf{r}}$. Jedinou výjimkou je matice úhlové rychlosti, která bude značena symbolem $\boldsymbol{\Omega}$. Platí ovšem, že $\boldsymbol{\Omega}$ je rovna $\tilde{\boldsymbol{\omega}}$.

Vektory ve formě antisymetrické matice jsou antikomutativní, tzn. splňují rovnici:

$${}^j\tilde{\boldsymbol{\omega}} {}^j\mathbf{r} = -{}^j\tilde{\mathbf{r}} {}^j\boldsymbol{\omega} \quad (2.13)$$

Důležitou poznámkou je, že antisymetrické matice se transformují jako tenzory druhého řádu [23, s. 263]:

$${}^i\tilde{\mathbf{r}} = \mathbf{S}_{ij} {}^j\tilde{\mathbf{r}} \mathbf{S}_{ij}^T \quad (2.14)$$

2.1.7 Cardanovy úhly

Pro popis natočení tělesa v prostoru jsou v této práci použity Cardanovy úhly. Jsou to 3 skalární veličiny ϕ_x , ϕ_y , ϕ_z pomocí kterých je počítána matice směrových cosinů. Matice směrových cosinů se pomocí Cardanových úhlů vypočítá jako:

$$\mathbf{S}_{ij} = \mathbf{S}_{\phi_x}(\phi_x)\mathbf{S}_{\phi_y}(\phi_y)\mathbf{S}_{\phi_z}(\phi_z) \quad (2.15)$$

Kde \mathbf{S}_{ϕ_k} je matice rotace okolo osy k . Takto definovanou maticí směrových cosinů lze reprezentovat jako transformaci mezi souřadnicovými systémy i a j tak, že jsou provedeny 3 za sebou jdoucí rotace, a to okolo osy X, Y, Z. Obdobným způsobem lze definovat i Cardanovy úhly pro sekvenci rotací např. okolo os Y, X, Z. V této práci bude využita právě sekvence X, Y, Z.

Matice rotace okolo jednotlivých os \mathbf{S}_{ϕ_k} jsou počítány jako:

$$\mathbf{S}_{\phi_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_x) & -\sin(\phi_x) \\ 0 & \sin(\phi_x) & \cos(\phi_x) \end{bmatrix} \quad (2.16)$$

$$\mathbf{S}_{\phi_y} = \begin{bmatrix} \cos(\phi_y) & 0 & \sin(\phi_y) \\ 0 & 1 & 0 \\ -\sin(\phi_y) & 0 & \cos(\phi_y) \end{bmatrix} \quad (2.17)$$

$$\mathbf{S}_{\phi_z} = \begin{bmatrix} \cos(\phi_z) & -\sin(\phi_z) & 0 \\ \sin(\phi_z) & \cos(\phi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.18)$$

2.2 Pohybové rovnice tělesa bez vazeb

V této sekci bude popsáno odvození pohybových rovnic pro tělesa bez vazeb.

2.2.1 Odvození hybnosti jednoho tělesa

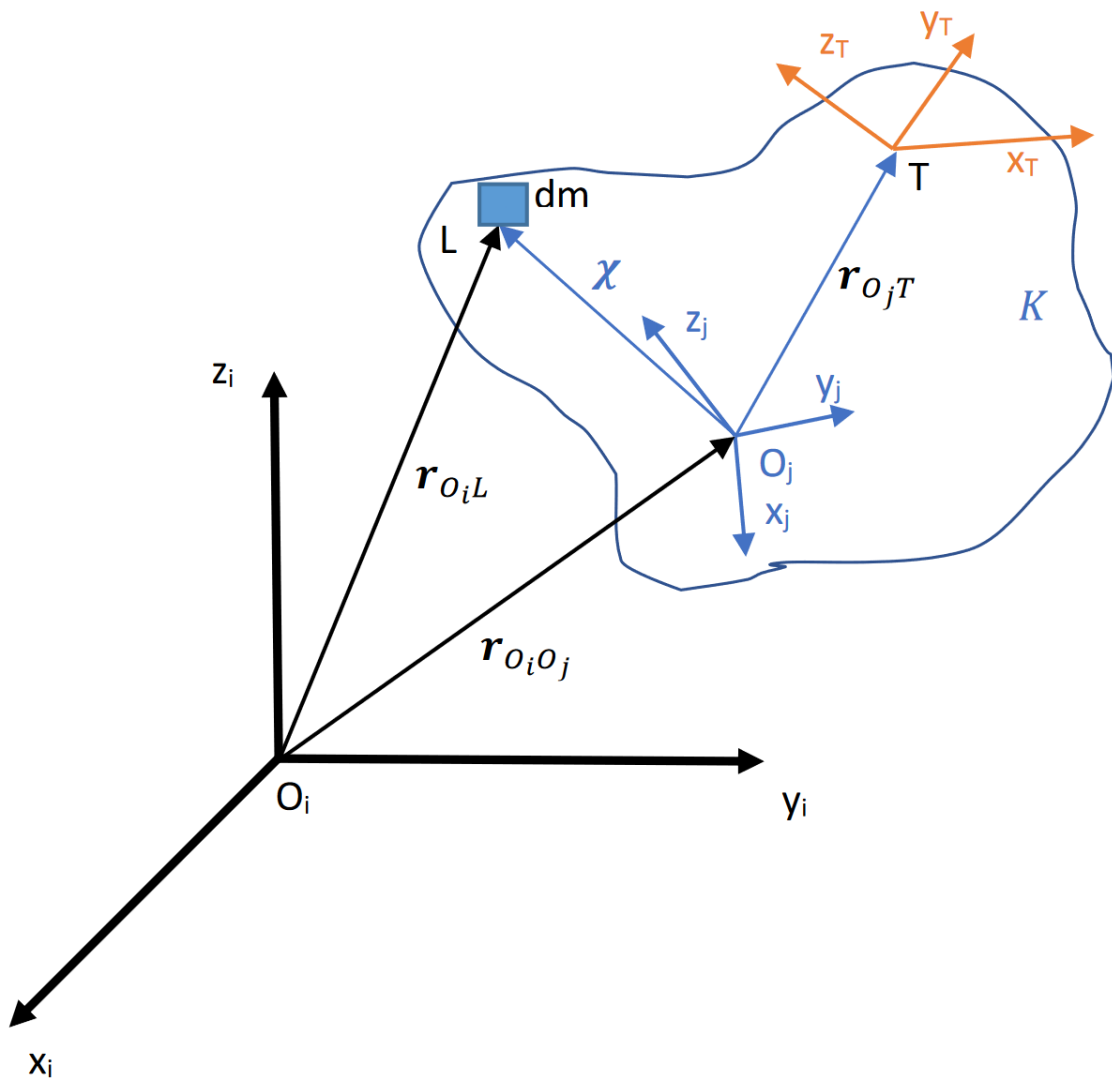
Pro odvození hybnosti jednoho tělesa se vychází z obr. 2.4. Na něm je znázorněno absolutně tuhé těleso označené písmenem K . Dále je zde souřadnicový systém i , ze kterého vede rádiusvektor $\mathbf{r}_{O_i O_j}$ spojující souřadnicový systém i s libovolným bodem na vyšetřovaném tělese O_j , odlišným od středu hmotnosti. Dále je na tělese zobrazen infinitesimálně velký element hmoty označený jako dm . Bod L je s počátkem O_j spojen vektorem $\boldsymbol{\chi}$. Posledním bodem na tělese je bod T, který znázorňuje střed hmotnosti tělesa.

K získání celkové hybnosti tělesa K , vzhledem k počátku souřadnicového systému i , je potřeba počítat (zintegrovat) všechny infinitezimální elementy hmoty vynásobené jejich rychlostí. Odvozená hybnost bude vyjádřena v souřadnicovém systému i .

$${}^i \mathbf{P} = \int {}^i \dot{\mathbf{r}}_{O_i L} dm \quad (2.19)$$

Integrál v rovnici (2.19) je integrálem přes celé těleso K .

Vektor ${}^i \mathbf{r}_{O_i L}$ lze zapsat jako součet vektoru ${}^i \mathbf{r}_{O_i O_j}$ a vektoru ${}^i \boldsymbol{\chi}$. Vektor ${}^i \boldsymbol{\chi}$ se dále vyjádří v souřadnicovém systému j :



Obr. 2.4: Vektorový diagram pro odvození hybnosti a momentu hybnosti

$${}^i \mathbf{r}_{O_i L} = {}^i \mathbf{r}_{O_i O_j} + {}^i \boldsymbol{\chi} = {}^i \mathbf{r}_{O_i O_j} + \mathbf{S}_{ij} {}^j \boldsymbol{\chi} \quad (2.20)$$

Časovou derivací rovnice (2.20) se získá:

$${}^i \dot{\mathbf{r}}_{O_i L} = {}^i \dot{\mathbf{r}}_{O_i O_j} + \dot{\mathbf{S}}_{ij} {}^j \boldsymbol{\chi} + \underbrace{\mathbf{S}_{ij} {}^j \dot{\boldsymbol{\chi}}}_{=0, \text{ protože } {}^j \dot{\boldsymbol{\chi}} = 0} \quad (2.21)$$

Poslední člen je roven nule z důvodu předpokladu absolutně tuhého tělesa. Časovou derivaci matice směrových cosinů lze dále rozepsat za využití matice úhlové rychlosti (viz subsekcce 2.1.5) jako:

$${}^i \dot{\mathbf{r}}_{O_i L} = {}^i \dot{\mathbf{r}}_{O_i O_j} + \mathbf{S}_{ij} {}^j \boldsymbol{\Omega}_{ij} {}^j \boldsymbol{\chi} \quad (2.22)$$

Dosazením rovnice (2.22) do (2.19) se získá:

$${}^i \mathbf{P} = \int ({}^i \dot{\mathbf{r}}_{O_i O_j} + \mathbf{S}_{ij} {}^j \boldsymbol{\Omega}_{ij} {}^j \boldsymbol{\chi}) dm = {}^i \dot{\mathbf{r}}_{O_i O_j} \int dm + \mathbf{S}_{ij} {}^j \boldsymbol{\Omega}_{ij} \int {}^j \boldsymbol{\chi} dm \quad (2.23)$$

Kde oba integrály jsou v čase konstantní. První část je integrál hmotnosti přes celé těleso. Výsledkem je celková hmotnost tělesa.

$$\int dm = m \quad (2.24)$$

Kde m je hmotnost celého tělesa.

Druhý integrál je statický moment tělesa, který lze vyjádřit pomocí hmotnosti tělesa a vzdálenosti bodu O_j a středu hmotnosti T [24, s. 67].

$$\int {}^j \boldsymbol{\chi} dm = m {}^j \mathbf{r}_{O_j T} \quad (2.25)$$

Po integraci a úpravě rovnice (2.23) přejde do tvaru:

$${}^i \mathbf{P} = m {}^i \dot{\mathbf{r}}_{O_i L} + m \mathbf{S}_{ij} {}^j \boldsymbol{\Omega}_{ij} {}^j \mathbf{r}_{TL} \quad (2.26)$$

Zvláštním případem je volba $O_j = T$, kdy zvolený bod splyne se středem hmoty tělesa. Rádusvektor ${}^j \mathbf{r}_{TL}$ je nulový a rovnice (2.26) přejde do tvaru:

$${}^i \mathbf{P} = m {}^i \dot{\mathbf{r}}_{O_i T} \quad (2.27)$$

2.2.2 Odvození momentu hybnosti tělesa

Pro odvození momentu hybnosti ${}^i \mathbf{L}_{O_i}$ tělesa K vůči počátku souřadnicového systému i se vychází z obr. 2.4. Celkový moment hybnosti lze vypočítat zintegrováním momentu hybnosti infinitezimálních elementů hmoty celého tělesa:

$${}^i\mathbf{L}_{O_i} = \int {}^i\tilde{\mathbf{r}}_{O_iL} {}^i\dot{\mathbf{r}}_{O_iL} dm \quad (2.28)$$

Kde ${}^i\tilde{\mathbf{r}}_{O_iL}$ je rádiusvektor spojující počátek souřadnicového systému i ve formě antisymetrické matice. (viz subsekcce 2.1.6)

Dosazením rovnice (2.20) do (2.28) vznikne:

$${}^i\mathbf{L}_{O_i} = \int ({}^i\tilde{\mathbf{r}}_{O_iO_j} + {}^i\tilde{\boldsymbol{\chi}})({}^i\dot{\mathbf{r}}_{O_iO_j} + {}^i\dot{\boldsymbol{\chi}}) dm \quad (2.29)$$

Roznásobením závorek vzniknou 4 separátní členy označené jako a,b,c,d

$${}^i\mathbf{L}_{O_i} = \int \underbrace{({}^i\tilde{\mathbf{r}}_{O_iO_j} {}^i\dot{\mathbf{r}}_{O_iO_j})}_a + \underbrace{({}^i\tilde{\boldsymbol{\chi}} {}^i\dot{\boldsymbol{\chi}})}_b + \underbrace{({}^i\tilde{\boldsymbol{\chi}} {}^i\dot{\mathbf{r}}_{O_iO_j})}_c + \underbrace{({}^i\tilde{\mathbf{r}}_{O_iO_j} {}^i\dot{\boldsymbol{\chi}})}_d dm \quad (2.30)$$

ad a:

Oba členy nejsou na integrovaném elementu závislé. Integrál lze přímo vypočítat jako:

$$\int {}^i\tilde{\mathbf{r}}_{O_iO_j} {}^i\dot{\mathbf{r}}_{O_iO_j} dm = m {}^i\tilde{\mathbf{r}}_{O_iO_j} {}^i\dot{\mathbf{r}}_{O_iO_j} \quad (2.31)$$

ad b:

Matice ${}^i\tilde{\boldsymbol{\chi}}$ je transformována dle (2.14) a vektor ${}^i\dot{\boldsymbol{\chi}}$ použitím rovnice (2.5):

$$\int {}^i\tilde{\boldsymbol{\chi}} {}^i\dot{\boldsymbol{\chi}} dm = \int \mathbf{S}_{ij} {}^j\tilde{\boldsymbol{\chi}} \mathbf{S}_{ij}^T ({}^j\dot{\boldsymbol{\chi}}) dm = \left(\int \mathbf{S}_{ij} {}^j\tilde{\boldsymbol{\chi}} \mathbf{S}_{ij}^T \mathbf{S}_{ij} {}^j\boldsymbol{\Omega}_{ij} {}^j\boldsymbol{\chi} dm \right) \quad (2.32)$$

Užitím vlastnosti vektorového součinu ${}^j\boldsymbol{\Omega}^j\boldsymbol{\chi} = -{}^j\tilde{\boldsymbol{\chi}} {}^j\boldsymbol{\omega}$ lze upravit rovnici (2.32):

$$\int {}^i\tilde{\boldsymbol{\chi}} {}^i\dot{\boldsymbol{\chi}} dm = \mathbf{S}_{ij} \underbrace{\left(- \int {}^j\tilde{\boldsymbol{\chi}} {}^j\boldsymbol{\chi} dm \right)}_{= {}^j\mathbf{J}_{O_j}} {}^j\boldsymbol{\omega}_{ij} = \mathbf{S}_{ij} {}^j\mathbf{J}_{O_j} {}^j\boldsymbol{\omega}_{ij} \quad (2.33)$$

Kde ${}^j\mathbf{J}_{O_j}$ je tenzor setrvačnosti tělesa vztažený k bodu O_j vyjádřený v souřadnicovém systému j . Vzhledem k tomu, že je souřadnicový systém j pevně spojen s tělesem, je tenzor setrvačnosti v čase konstantní. Jednotlivé složky tenzoru se vypočítají jako:

$$\begin{aligned}
{}^j\mathbf{J}_{O_j} &= - \int {}^j\tilde{\boldsymbol{\chi}} \quad {}^j\tilde{\boldsymbol{\chi}} dm \\
&= \int \begin{bmatrix} 0 & {}^iz_{O_iT} & -{}^iy_{O_iT} \\ -{}^iz_{O_iT} & 0 & {}^ix_{O_iT} \\ {}^iy_{O_iT} & -{}^ix_{O_iT} & 0 \end{bmatrix} \begin{bmatrix} 0 & -{}^iz_{O_iT} & {}^iy_{O_iT} \\ {}^iz_{O_iT} & 0 & -{}^ix_{O_iT} \\ -{}^iy_{O_iT} & {}^ix_{O_iT} & 0 \end{bmatrix} dm \\
&= \int \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} dm
\end{aligned} \tag{2.34}$$

ad c:

Část c lze upravit pomocí rovnice (2.25):

$$\int {}^i\tilde{\boldsymbol{\chi}} \quad {}^i\dot{\mathbf{r}}_{O_iO_j} dm = \left(\int {}^i\tilde{\boldsymbol{\chi}} dm \right) \quad {}^i\dot{\mathbf{r}}_{O_iO_j} = m \quad {}^i\tilde{\mathbf{r}}_{O_jT} \quad {}^i\dot{\mathbf{r}}_{O_iO_j} \tag{2.35}$$

ad d:

V poslední části je třeba vektor $\dot{\boldsymbol{\chi}}$ vyjádřit v souřadnicovém systému j .

$${}^i\dot{\boldsymbol{\chi}} = \mathbf{S}_{ij} \quad {}^j\boldsymbol{\Omega}_{ij} \quad {}^j\boldsymbol{\chi} \tag{2.36}$$

Spolu s využitím rovnice (2.25):

$$\int ({}^i\tilde{\mathbf{r}}_{O_iO_j} \quad {}^i\dot{\boldsymbol{\chi}}) dm = {}^i\tilde{\mathbf{r}}_{O_iO_j} \left(\int {}^i\dot{\boldsymbol{\chi}} dm \right) = {}^i\tilde{\mathbf{r}}_{O_iO_j} \mathbf{S}_{ij} \quad {}^j\boldsymbol{\Omega}_{ij} \left(\int {}^j\boldsymbol{\chi} dm \right) \tag{2.37}$$

Výsledný tvar:

$$\int ({}^i\tilde{\mathbf{r}}_{O_iO_j} \quad {}^i\dot{\boldsymbol{\chi}}) dm = m \quad {}^i\tilde{\mathbf{r}}_{O_iO_j} \mathbf{S}_{ij} \quad {}^j\boldsymbol{\Omega}_{ij} \quad {}^j\mathbf{r}_{O_jT} \tag{2.38}$$

Následně je třeba všechny dílčí výsledky bodů a,b,c,d spojit do výsledného momentu hybnosti:

$${}^i\mathbf{L}_{O_i} = m \quad {}^i\tilde{\mathbf{r}}_{O_iO_j} \quad {}^i\dot{\mathbf{r}}_{O_iO_j} + \mathbf{S}_{ij} \quad {}^j\mathbf{J}_{O_j} \quad {}^j\boldsymbol{\omega}_{ij} + \tag{2.39}$$

$$m \quad {}^i\tilde{\mathbf{r}}_{O_jT} \quad {}^i\dot{\mathbf{r}}_{O_iO_j} + m \quad {}^i\tilde{\mathbf{r}}_{O_iO_j} \mathbf{S}_{ij} \quad {}^j\boldsymbol{\Omega}_{ij} \quad {}^j\mathbf{r}_{O_jT} \tag{2.40}$$

Při výběru bodu $O_j = T$, se moment hybnosti zjednodušuje na

$${}^i\mathbf{L}_{O_i} = m \quad {}^i\tilde{\mathbf{r}}_{O_jT} \quad {}^i\dot{\mathbf{r}}_{O_iO_j} + \mathbf{S}_{ij} \quad {}^j\mathbf{J}_{O_j} \quad {}^j\boldsymbol{\omega}_{ij} \tag{2.41}$$

2.2.3 Newton - Eulerovy rovnice

Newtonov-Eulerovy rovnice budou odvozeny ze zákona zachování hybnosti a momentu hybnosti. Zvolený souřadnicový systém, ve kterém budou pohybové rovnice odvozeny je systém rámu, dále označovaný číslem 1.

$$\frac{d}{dt} {}^1\mathbf{P} = \sum_k {}^1\mathbf{F}_k \quad (2.42)$$

$$\frac{d}{dt} {}^1\mathbf{L}_{O_1} = \sum_k {}^1\mathbf{M}_{O_1k} \quad (2.43)$$

- ${}^1\mathbf{P}$ je hybnost tělesa vyjádřená v souřadnicovém systému rámu
- ${}^1\mathbf{L}_{O_1}$ je moment hybnosti tělesa vyjádřený v souřadnicovém systému rámu vzhledem k jeho počátku
- $\sum {}^1\mathbf{F}_k$ je suma všech vnějších sil působících na těleso vyjádřených v souřadnicovém systému rámu
- $\sum {}^1\mathbf{M}_{O_1k}$ je součet momentů sil vzhledem k počátku souřadnicového systému rámu působící na těleso vyjádřených v souřadnicovém systému rámu.

Pohybové rovnice budou odvozeny s předpokladem, že zvolený bod na tělese bude shodný se středem hmotnosti tělesa tzn. $O_j = T$. Tento předpoklad velmi zjednoduší další odvození. V praxi to znamená, že na místo složitých výrazů pro získání hybnosti a momentu hybnosti (rovnice (2.19), (2.40)) je možné použít jejich jednodušší formy (rovnice (2.27), (2.41)).

Prvním krokem je do rovnice (2.42) dosadit vztah pro hybnost (2.27).

$$\frac{d}{dt} {}^1\mathbf{P} = \frac{d}{dt} (m {}^1\dot{\mathbf{r}}_{O_1T}) = m {}^1\ddot{\mathbf{r}}_{O_1T} = \sum {}^1\mathbf{F}_k \quad (2.44)$$

Rovnice (2.44) je Newtonovou pohybovou rovnicí pro jedno těleso.

Druhým krokem je dosazení rovnice (2.41) do (2.43)

$$\frac{d}{dt} {}^1\mathbf{L}_{O_1} = \frac{d}{dt} (m {}^1\tilde{\mathbf{r}}_{O_1T} {}^1\dot{\mathbf{r}}_{O_1T} + \mathbf{S}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j}) \quad (2.45)$$

Po derivování jednotlivých členů se získá 5 členů rovnice:

$$\begin{aligned} \frac{d}{dt} {}^1\mathbf{L}_{O_1} = & \underbrace{m \dot{{}^1\tilde{\mathbf{r}}_{O_1T}} {}^1\dot{\mathbf{r}}_{O_1T}}_a + \underbrace{m {}^1\tilde{\mathbf{r}}_{O_1T} {}^1\ddot{\mathbf{r}}_{O_1T}}_b + \\ & \underbrace{\dot{\mathbf{S}}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j}}_c + \underbrace{\mathbf{S}_{1j} \dot{{}^j\mathbf{J}}_T {}^j\boldsymbol{\omega}_{1j}}_d + \underbrace{\mathbf{S}_{1j} {}^j\mathbf{J}_T {}^j\dot{\boldsymbol{\omega}}_{1j}}_e \end{aligned} \quad (2.46)$$

ad a:

Vektorový součin vektoru sám se sebou je vždy nula:

$$m \mathbf{}^1\dot{\mathbf{r}}_{O_1T} \mathbf{}^1\dot{\mathbf{r}}_{O_1T} = m \mathbf{}^1\dot{\mathbf{r}}_{O_1T} \times \mathbf{}^1\dot{\mathbf{r}}_{O_1T} = 0 \quad (2.47)$$

ad c:

Úprava derivace matice přechodu viz subsekce 2.1.5

$$\dot{\mathbf{S}}_{1j} \mathbf{}^j\mathbf{J}_T \mathbf{}^j\boldsymbol{\omega}_{1j} = \mathbf{S}_{1j} \mathbf{}^j\boldsymbol{\Omega}_{1j} \mathbf{}^j\mathbf{J}_T \mathbf{}^j\boldsymbol{\omega}_{1j} \quad (2.48)$$

ad d:

$$\mathbf{S}_{1j} \mathbf{}^j\dot{\mathbf{J}}_T \mathbf{}^j\boldsymbol{\omega}_{1j} = 0 \quad (2.49)$$

protože

$$\mathbf{}^j\dot{\mathbf{J}}_T = 0 \quad (2.50)$$

Vztah (2.50) platí, protože tenzor setrvačnosti vyjádřený v souřadnicovém systému tělesa vztažený k bodu pevně spojeného s tělesem je konstantní. Proto je časová derivace nulová.

Vztah pro derivaci momentu hybnosti (2.46) se pak zjednoduší do podoby:

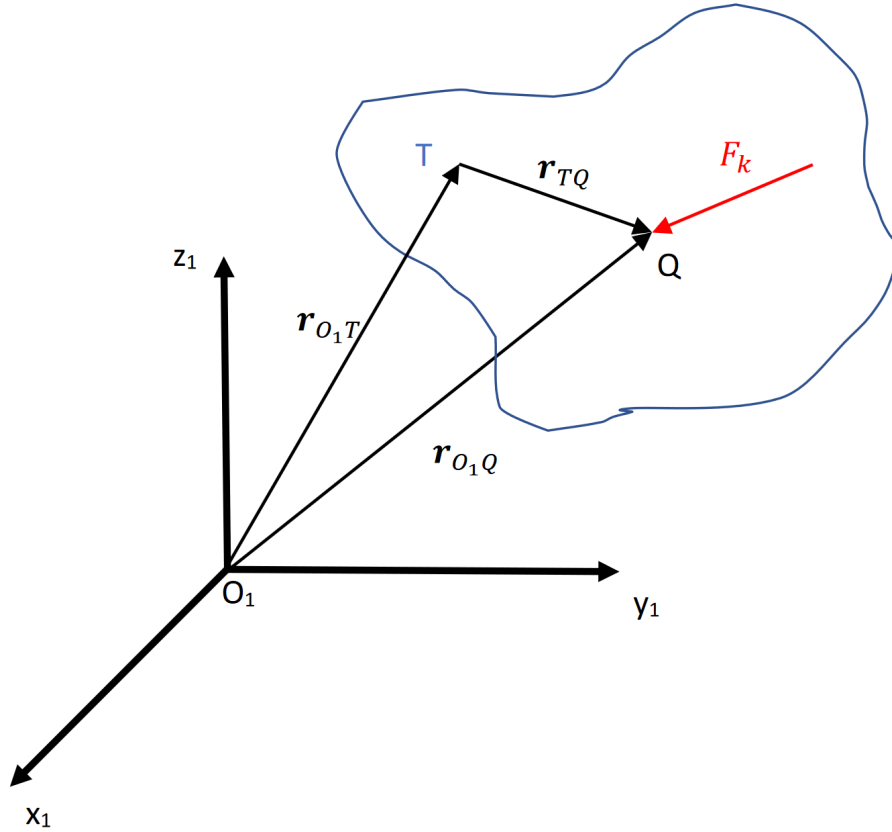
$$\frac{d}{dt} \mathbf{}^1\mathbf{L}_{O_1} = m \mathbf{}^1\tilde{\mathbf{r}}_{O_1T} \mathbf{}^1\ddot{\mathbf{r}}_{O_1T} + \mathbf{S}_{1j} \mathbf{}^j\boldsymbol{\Omega}_{1j} \mathbf{}^j\mathbf{J}_T \mathbf{}^j\boldsymbol{\omega}_{1j} + \mathbf{S}_{1j} \mathbf{}^j\mathbf{J}_T \mathbf{}^j\dot{\boldsymbol{\omega}}_{1j} = \sum_k \mathbf{}^1\mathbf{M}_{O_1k} \quad (2.51)$$

Momentové účinky od sil

Momenty síly $\mathbf{}^1\mathbf{M}_{O_1k}$ vystupující v rovnici (2.51) lze rozdělit na momentové účinky působící přímo na těleso a na momenty síly generované od lineárních sil.

$$\begin{aligned} \sum_k \mathbf{}^1\mathbf{M}_{O_1k} &= \sum_k \mathbf{}^1\mathbf{M}_{O_1Mk} + \sum_k \mathbf{}^1\tilde{\mathbf{r}}_{O_1Q} \mathbf{}^1\mathbf{F}_k \\ &= \sum_k \mathbf{}^1\mathbf{M}_{O_1Mk} + \sum_k (\mathbf{}^1\tilde{\mathbf{r}}_{O_1T} + \mathbf{}^1\tilde{\mathbf{r}}_{TQ}) \mathbf{}^1\mathbf{F}_k \\ &= \sum_k \mathbf{}^1\mathbf{M}_{O_1Mk} + \sum_k \mathbf{}^1\tilde{\mathbf{r}}_{TQ} \mathbf{}^1\mathbf{F}_k + \sum_k \mathbf{}^1\tilde{\mathbf{r}}_{O_1T} \mathbf{}^1\mathbf{F}_k \\ &= \sum_k \mathbf{}^1\mathbf{M}_{Tk} + \mathbf{}^1\tilde{\mathbf{r}}_{O_1T} \sum_k \mathbf{}^1\mathbf{F}_k \end{aligned} \quad (2.52)$$

Kde $\mathbf{}^1\mathbf{M}_{O_1Mk}$ jsou momenty sil působící přímo na těleso a součin $\mathbf{}^1\tilde{\mathbf{r}}_{O_1Q} \mathbf{}^1\mathbf{F}_k$ jsou momenty vyvolané působením lineárních sil na těleso. Vektor \mathbf{r}_{O_1Q} je podle obr. 2.5 možné rozdělit na součet vektoru mezi rámem a středem hmotnosti tělesa $\mathbf{}^1\mathbf{r}_{O_1T}$ s vektorem $\mathbf{}^1\mathbf{r}_{TQ}$ spojujícího střed hmotnosti s bodem, ve kterém síla \mathbf{F}_k působí. Proměnná $\mathbf{}^1\mathbf{M}_{Tk}$ souhrnně označuje momentové účinky od momentů sil a od sil vzhledem k hmotnému středu tělesa.



Obr. 2.5: Silový účinek na těleso

Dosazením rovnice (2.44) do (2.52) vznikne:

$$\sum_k {}^1\mathbf{M}_{O_1k} = \sum_k {}^1\mathbf{M}_{Tk} + m {}^1\tilde{\mathbf{r}}_{O_1T} {}^1\ddot{\mathbf{r}}_{O_1T} \quad (2.53)$$

Dosazením (2.53) do (2.51) lze vykrátit stejné výrazy na levé a pravé straně:

$$m {}^1\tilde{\mathbf{r}}_{O_1T} {}^1\ddot{\mathbf{r}}_{O_1T} + \mathbf{S}_{1j} {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} + \mathbf{S}_{1j} {}^j\mathbf{J}_T {}^j\dot{\boldsymbol{\omega}}_{1j} = \sum_k {}^1\mathbf{M}_{Tk} + m {}^1\tilde{\mathbf{r}}_{O_1T} {}^1\ddot{\mathbf{r}}_{O_1T} \quad (2.54)$$

Eliminace druhé časové derivace polohového vektoru ${}^1\tilde{\mathbf{r}}_{O_1T}$ z výrazu pro moment hybnosti byl hlavní cíl rozkladu momentu v (2.52). Díky tomuto kroku nejsou rovnice popisující lineární pohyb tělesa a rotační pohyb tělesa provázané.

Maticový tvar Newtonových-Eulerových rovnic

Přenásobením rovnice (2.54) inverzní maticí přechodu \mathbf{S}_{j1} vznikne:

$$\mathbf{S}_{j1} \mathbf{S}_{1j} {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} + \mathbf{S}_{j1} \mathbf{S}_{1j} {}^j\mathbf{J}_T {}^j\dot{\boldsymbol{\omega}}_{1j} = \mathbf{S}_{j1} \sum_k {}^1\mathbf{M}_{Tk} = \sum_k {}^j\mathbf{M}_{Tk} \quad (2.55)$$

$${}^j\mathbf{J}_T {}^j\dot{\boldsymbol{\omega}}_{1j} = \sum_k {}^j\mathbf{M}_{Tk} - {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} \quad (2.56)$$

Dohromady lze pohybové rovnice tělesa bez vazeb vzhledem k souřadnicovému systému rámu napsat v maticovém tvaru jako:

$$\begin{bmatrix} m\mathbf{E}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & {}^j\mathbf{J}_T \end{bmatrix} \begin{bmatrix} {}^1\ddot{\mathbf{r}}_{O_1T} \\ {}^j\dot{\boldsymbol{\omega}}_{1j} \end{bmatrix} = \begin{bmatrix} \sum_k {}^1\mathbf{F}_k \\ - {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} + \sum_k {}^j\mathbf{M}_{Tk} \end{bmatrix} \quad (2.57)$$

Případně v symbolické formě jako:

$$\mathbf{M}(\mathbf{s})\dot{\mathbf{v}} = \mathbf{f} + \mathbf{q}_G \quad (2.58)$$

Kde \mathbf{f} je vektor vnějších sil a momentů působících na těleso

$$\mathbf{f} = \begin{bmatrix} \sum_k {}^1\mathbf{F}_k \\ \sum_k {}^j\mathbf{M}_{Tk} \end{bmatrix} \quad (2.59)$$

A \mathbf{q}_G je vektor gyroskopických silových účinků

$$\mathbf{q}_G = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ - {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} \end{bmatrix} \quad (2.60)$$

Je důležité si povšimnout, že zrychlení ${}^1\ddot{\mathbf{r}}_{O_1T}$ a ${}^j\dot{\boldsymbol{\omega}}_{1j}$ jsou psané v jiných souřadnicových systémech. Poloha hmotného středu tělesa je vyjádřena v souřadnicovém systému rámu, zatímco úhlová rychlost je vyjádřena v souřadnicovém systému tělesa. Pro přepočítání do souřadnicového systému tělesa stačí výslednou úhlovou rychlost vynásobit maticí přechodu od báze tělesa k bázi rámu \mathbf{S}_{1j} .

Pokud by byly pohybové rovnice psány tak, aby úhlová rychlost byla přímo vyjádřena v souřadnicovém systému rámu, rovnice by se staly výpočetně náročnější. V každém integračním kroku by bylo třeba transformovat tenzor setrvačnosti do souřadnicového systému rámu. Hlavní nevýhodou by byla nekonstantní matice hmotnosti systému. Výsledné rovnice by vypadaly následovně:

$$\begin{bmatrix} m\mathbf{E}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{S}_{1j} {}^j\mathbf{J}_T \mathbf{S}_{j1} \end{bmatrix} \begin{bmatrix} {}^1\ddot{\mathbf{r}}_{O_1T} \\ {}^1\dot{\boldsymbol{\omega}}_{1j} \end{bmatrix} = \begin{bmatrix} \sum_k {}^1\mathbf{F}_k \\ - \mathbf{S}_{1j} {}^j\boldsymbol{\Omega}_{1j} {}^j\mathbf{J}_T {}^j\boldsymbol{\omega}_{1j} + \sum_k {}^1\mathbf{M}_{Tk} \end{bmatrix} \quad (2.61)$$

Tento tvar rovnic ale v této práci použit nebude.

V neposlední řadě nebyl ještě diskutován vliv volby hmotného středu tělesa jako vyšetřovaného bodu na tělese. Tato volba přinesla značné zjednodušení v rovnicích (2.27) a (2.41). Kromě zjednodušení odvození tato volba přinesla výhodu ve formě zrušení provázání lineární a úhlové rychlosti. Při volbě $O_j \neq T$ mimodiagonální členy v matici (2.57) nejsou nulové a rovnice se stávají opět výpočetně náročnější. Obecné odvození je možné najít v [23, s. 129-151].

2.3 Kinematické rovnice

Pro popis konfigurace soustavy těles je potřeba zavést vektor všech fyzikálních souřadnic \mathbf{s} . Ten je složen z vektorů fyzikálních souřadnic jednotlivých těles \mathbf{s}_j , které se dále dělí na souřadnice translační uložené ve vektorech ${}^1\mathbf{r}_j$ a úhlové souřadnice uložené ve vektoru $\boldsymbol{\eta}_j$. Jako úhlové souřadnice jsou této práci použity Cardanovy úhly.

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_{n_b} \end{bmatrix} \quad \mathbf{s}_j = \begin{bmatrix} {}^1\mathbf{r}_j \\ \boldsymbol{\eta}_j \end{bmatrix} \quad {}^1\mathbf{r}_j = \begin{bmatrix} {}^1x_j \\ {}^1y_j \\ {}^1z_j \end{bmatrix} \quad \boldsymbol{\eta}_j = \begin{bmatrix} \phi_{xj} \\ \phi_{yj} \\ \phi_{zj} \end{bmatrix} \quad (2.62)$$

Dále je třeba zavést vektor časové derivace $\dot{\mathbf{s}}$ vektoru souřadnic:

$$\dot{\mathbf{s}} = \begin{bmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \\ \vdots \\ \dot{\mathbf{s}}_{n_b} \end{bmatrix} \quad \dot{\mathbf{s}}_j = \begin{bmatrix} {}^1\dot{\mathbf{r}}_j \\ \dot{\boldsymbol{\eta}}_j \end{bmatrix} \quad {}^1\dot{\mathbf{r}}_j = \begin{bmatrix} {}^1\dot{x}_j \\ {}^1\dot{y}_j \\ {}^1\dot{z}_j \end{bmatrix} \quad \dot{\boldsymbol{\eta}}_j = \begin{bmatrix} \dot{\phi}_{xj} \\ \dot{\phi}_{yj} \\ \dot{\phi}_{zj} \end{bmatrix} \quad (2.63)$$

A zavést vektor rychlostí \mathbf{v} jako:

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_{n_c} \end{bmatrix} \quad \mathbf{v}_j = \begin{bmatrix} {}^1\dot{\mathbf{r}}_j \\ {}^j\boldsymbol{\omega}_{1j} \end{bmatrix} \quad {}^j\boldsymbol{\omega}_{1j} = \begin{bmatrix} {}^j\omega_{xj} \\ {}^j\omega_{yj} \\ {}^j\omega_{zj} \end{bmatrix} \quad (2.64)$$

2.3.1 Přepočítání vektoru rychlostí na derivaci vektoru souřadnic

Hlavní rozdíl mezi modelem dynamiky 2D a 3D těles je jejich popis dynamiky rotace. Zatímco u 2D těles je derivace souřadnic $\dot{\mathbf{s}}$ a vektor rychlostí \mathbf{v} identický, u 3D úloh tomu tak není. Je potřeba odvodit vztah mezi těmito dvěma vektory pro další výpočty.

Vychází se z výpočtu matice přechodu dané Cardanovými úhly definované v subsekcí (2.15):

$$\mathbf{S}_{ij} = \mathbf{S}_{\phi_x}(\phi_x)\mathbf{S}_{\phi_y}(\phi_y)\mathbf{S}_{\phi_z}(\phi_z) \quad (2.65)$$

Následně se provede derivace matice přechodu:

$$\dot{\mathbf{S}}_{ij} = \frac{d}{dt} \left(\mathbf{S}_{\phi_x}(\phi_x)\mathbf{S}_{\phi_y}(\phi_y)\mathbf{S}_{\phi_z}(\phi_z) \right) = \mathbf{S}_{ij} {}^j\boldsymbol{\Omega}_{ij} \quad (2.66)$$

A je možné vyjádřit matici úhlové rychlosti ${}^j\boldsymbol{\Omega}_{ij}$:

$${}^j\boldsymbol{\Omega}_{ij} = \mathbf{S}_{ij}^T \dot{\mathbf{S}}_{ij} \quad (2.67)$$

Převodem antisymetrické matice do vektorového tvaru vznikne vektor úhlové rychlosti s následujícími složkami:

$$\begin{aligned} {}^j\boldsymbol{\omega}_{ij} &= \begin{bmatrix} \cos(\phi_y) \cos(\phi_z) \dot{\phi}_x + \sin(\phi_z) \dot{\phi}_y \\ -\cos(\phi_y) \sin(\phi_z) \dot{\phi}_x + \cos(\phi_z) \dot{\phi}_y \\ \sin(\phi_y) \dot{\phi}_x + \dot{\phi}_z \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi_y) \cos(\phi_z) & \sin(\phi_z) & 0 \\ -\cos(\phi_y) \sin(\phi_z) & \cos(\phi_z) & 0 \\ \sin(\phi_y) & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ \dot{\phi}_z \end{bmatrix}}_{\dot{\boldsymbol{\eta}}} \end{aligned} \quad (2.68)$$

Přenásobením maticí transformace \mathbf{S}_{ij} vznikne vztah pro definici matice \mathbf{H}^{-1} , která slouží pro přepočítání derivace vektoru Cardanových úhlů $\dot{\boldsymbol{\eta}}$ na úhlovou rychlost ${}^j\boldsymbol{\omega}_{ij}$:

$$\mathbf{S}_{ij} {}^j\boldsymbol{\omega}_{ij} = {}^i\boldsymbol{\omega}_{ij} = \underbrace{\begin{bmatrix} 1 & 0 & \sin(\phi_y) \\ 0 & \cos(\phi_x) & -\sin(\phi_x) \cos(\phi_y) \\ 0 & \sin(\phi_x) & \cos(\phi_x) \cos(\phi_y) \end{bmatrix}}_{\mathbf{H}^{-1}(\boldsymbol{\eta})} \underbrace{\begin{bmatrix} \dot{\phi}_x \\ \dot{\phi}_y \\ \dot{\phi}_z \end{bmatrix}}_{\dot{\boldsymbol{\eta}}} \quad (2.69)$$

Matice $\mathbf{H}(\boldsymbol{\eta})$ pak vypadá následovně:

$$\mathbf{H}(\boldsymbol{\eta}) = \frac{1}{\cos(\phi_y)} \begin{bmatrix} \cos(\phi_y) & \sin(\phi_x) \sin(\phi_y) & -\cos(\phi_x) \sin(\phi_y) \\ 0 & \cos(\phi_x) \cos(\phi_y) & \sin(\phi_x) \cos(\phi_y) \\ 0 & -\sin(\phi_x) & \cos(\phi_y) \end{bmatrix} \quad (2.70)$$

Platí vztahy:

$$\dot{\boldsymbol{\eta}} = \mathbf{H}(\boldsymbol{\eta}) {}^i\boldsymbol{\omega}_{ij} = \mathbf{H}(\boldsymbol{\eta}) \mathbf{S}_{ij} {}^j\boldsymbol{\omega}_{ij} \quad (2.71)$$

$${}^j\boldsymbol{\omega}_{ij} = \mathbf{S}_{ji} \mathbf{H}^{-1}(\boldsymbol{\eta}) \dot{\boldsymbol{\eta}} \quad (2.72)$$

Dohromady lze zapsat derivaci polohového vektoru ${}^1\mathbf{r}$ a vektoru Cardanových úhlů $\dot{\boldsymbol{\eta}}$ jako:

$$\dot{\mathbf{s}} = \begin{bmatrix} {}^1\dot{\mathbf{r}}_j \\ \dot{\boldsymbol{\eta}}_j \end{bmatrix} = \begin{bmatrix} \mathbf{E}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{H}(\boldsymbol{\eta}) \mathbf{S}_{ij} \end{bmatrix} \begin{bmatrix} {}^1\mathbf{r} \\ {}^j\boldsymbol{\omega} \end{bmatrix} = \mathbf{T}(\mathbf{s}) \mathbf{v} \quad (2.73)$$

Matice přepočtu vektoru rychlostí \mathbf{v} na derivaci vektoru souřadnic $\dot{\mathbf{s}}$ je defino-

vána jako:

$$\mathbf{T}(\mathbf{s}) = \begin{bmatrix} \mathbf{E}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{H}(\boldsymbol{\eta})\mathbf{S}_{ij} \end{bmatrix} \quad (2.74)$$

2.3.2 Rovnice kinematiky

Pokud modelovaná soustava těles obsahuje vazby mezi tělesy, pak jsou vazby modelovány za pomoci n_c nezávislých vazbových rovnic. Vazbové rovnice jsou zapísány ve tvaru:

$$\mathbf{g}(\mathbf{s}) = \mathbf{0} \quad (2.75)$$

Jeich časová derivace, která musí také zůstat nulová, se vypočte jako:

$$\dot{\mathbf{g}}(\mathbf{s}) = \frac{\partial \mathbf{g}(\mathbf{s})}{\partial \mathbf{s}} \dot{\mathbf{s}} = \mathbf{g}_s(\mathbf{s}) \dot{\mathbf{s}} = \mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s}) \mathbf{v} = \mathbf{0} \quad (2.76)$$

Kde do rovnice (2.76) byla dosazena rovnice (2.73)

Stejně tak i druhá časová derivace musí zůstat nulová:

$$\ddot{\mathbf{g}}(\mathbf{s}) = \frac{\partial}{\partial \mathbf{s}} (\mathbf{g}_s(\mathbf{s}) \dot{\mathbf{s}}) \dot{\mathbf{s}} + \mathbf{g}_s(\mathbf{s}) \ddot{\mathbf{s}} = \mathbf{0} \quad (2.77)$$

Časovou derivací rovnice (2.73) se získá:

$$\ddot{\mathbf{s}} = \dot{\mathbf{T}}\mathbf{v} + \mathbf{T}\dot{\mathbf{v}} \quad (2.78)$$

Dosazením rovnic (2.73) a (2.78) do rovnice (2.77) vznikne:

$$\ddot{\mathbf{g}}(\mathbf{s}) = \frac{\partial}{\partial \mathbf{s}} (\mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s}) \mathbf{v}) \mathbf{T}(\mathbf{s}) \mathbf{v} + \mathbf{g}_s(\mathbf{s}) (\dot{\mathbf{T}}\mathbf{v} + \mathbf{T}\dot{\mathbf{v}}) = \mathbf{0} \quad (2.79)$$

Vyjádřením výrazu $\mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s}) \dot{\mathbf{v}}$ lze definovat vektor pravé strany $\boldsymbol{\beta}_c(\mathbf{s}, \dot{\mathbf{s}})$ jako:

$$\mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s}) \dot{\mathbf{v}} = -\frac{\partial}{\partial \mathbf{s}} (\mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s}) \mathbf{v}) \mathbf{T}(\mathbf{s}) \mathbf{v} - \mathbf{g}_s(\mathbf{s}) (\dot{\mathbf{T}}\mathbf{v}) \equiv \boldsymbol{\beta}_c(\mathbf{s}, \dot{\mathbf{s}}) \quad (2.80)$$

Způsob výpočtu matice $\mathbf{g}_s(\mathbf{s}) \mathbf{T}(\mathbf{s})$ a vektoru pravé strany $\boldsymbol{\beta}_c(\mathbf{s}, \dot{\mathbf{s}})$ pro sférickou vazbu bude podrobněji popsán v sekci 2.9.

2.4 Vazbové reakční síly

Virtuální posunutí $\delta \mathbf{r}_L$ systému je definováno jako infinitezimálně malá změna polohy bodu \mathbf{r}_L za konstantního času t a dodržení vazbových podmínek. Virtuální práce vykonaná vazbovými reakcemi ${}^c \bar{\mathbf{f}}$ je nulová [23, s. 121]:

$$\mathbf{W} = {}^c\bar{\mathbf{f}}^T \delta \mathbf{s} = 0 \quad (2.81)$$

Vektor reakčních sil ${}^c\bar{\mathbf{f}}$ působí na souřadnicích \mathbf{s}

$${}^c\bar{\mathbf{f}} = \begin{bmatrix} {}^c\bar{\mathbf{f}}_1 \\ {}^c\bar{\mathbf{f}}_2 \\ \vdots \\ {}^c\bar{\mathbf{f}}_{n_b} \end{bmatrix} \quad {}^c\bar{\mathbf{f}}_i = \begin{bmatrix} {}^1f_{xi} \\ {}^1f_{yi} \\ {}^1f_{zi} \\ M_{xi} \\ M_{yi} \\ M_{zi} \end{bmatrix} \quad (2.82)$$

Předpokládá se, že vazbové podmínky jsou mezi sebou lineárně nezávislé. V každém časovém okamžiku má Jacobiho matice vazbových rovnic plnou hodnost:

$$\text{rank} \left(\frac{\partial \mathbf{g}(\mathbf{s})}{\partial \mathbf{s}} \right) = \text{rank}(\mathbf{g}_s(\mathbf{s})) = n_c \quad (2.83)$$

Vazbové rovnice lze rozvinout do Taylorovy řady prvního řádu:

$$\mathbf{g}(\mathbf{s} + \delta \mathbf{s}) = \mathbf{g}(\mathbf{s}) + \mathbf{g}_s(\mathbf{s}) \delta \mathbf{s} \quad (2.84)$$

A protože platí vazbové rovnice:

$$\mathbf{g}(\mathbf{s}) = \mathbf{0} \quad (2.85)$$

I při virtuálním posunu z definice:

$$\mathbf{g}(\mathbf{s} + \delta \mathbf{s}) = \mathbf{0} \quad (2.86)$$

Pak z rovnice (2.84) plyne nutnost splnit vztah:

$$\mathbf{g}_s(\mathbf{s}) \delta \mathbf{s} = \mathbf{0} \quad (2.87)$$

Souřadnice z vektoru všech souřadnic \mathbf{s} lze rozdělit na n_c závislých souřadnic $\mathbf{z} \in \mathbb{R}^{n_c}$ a $6n_b - n_c$ nezávislých souřadnic $\mathbf{q} \in \mathbb{R}^{6n_b - n_c}$ a obdobným způsobem rozdělit i Jacobiho matici $\mathbf{g}_s(\mathbf{s})$

$$\mathbf{s} = \begin{bmatrix} \mathbf{z} \\ \mathbf{q} \end{bmatrix} \quad \mathbf{g}_s = [\mathbf{g}_z, \mathbf{g}_q] \quad (2.88)$$

$$[\mathbf{g}_z, \mathbf{g}_q] \begin{bmatrix} \delta \mathbf{z} \\ \delta \mathbf{q} \end{bmatrix} = \mathbf{g}_z \delta \mathbf{z} + \mathbf{g}_q \delta \mathbf{q} = 0 \quad (2.89)$$

$$\mathbf{g}_z \delta \mathbf{z} = -\mathbf{g}_q \delta \mathbf{q} \quad (2.90)$$

Síly lze také rozdělit na síly působící na závislých ${}_z\mathbf{f}$ a nezávislých ${}_q\mathbf{f}$ souřadni-

cích:

$${}^c\bar{\mathbf{f}} = \begin{bmatrix} {}_z\bar{\mathbf{f}} \\ {}_q\bar{\mathbf{f}} \end{bmatrix} \quad (2.91)$$

$$\begin{bmatrix} {}_z\bar{\mathbf{f}}^T & {}_q\bar{\mathbf{f}}^T \end{bmatrix} \begin{bmatrix} \delta\mathbf{z} \\ \delta\mathbf{q} \end{bmatrix} = {}_z\bar{\mathbf{f}}^T \delta\mathbf{z} + {}_q\bar{\mathbf{f}}^T \delta\mathbf{q} = \mathbf{0} \quad (2.92)$$

$${}_z\bar{\mathbf{f}}^T \delta\mathbf{z} = -{}_q\bar{\mathbf{f}}^T \delta\mathbf{q} \quad (2.93)$$

Spojením rovnice (2.93) spolu s rovnicí (2.90) vznikne vektorová rovnice:

$$\begin{bmatrix} {}_z\bar{\mathbf{f}}^T \\ \mathbf{g}_z \end{bmatrix} \delta\mathbf{z} = - \begin{bmatrix} {}_q\bar{\mathbf{f}}^T \\ \mathbf{g}_q \end{bmatrix} \delta\mathbf{q} \quad (2.94)$$

Matice $\begin{bmatrix} {}_z\bar{\mathbf{f}}^T \\ \mathbf{g}_z \end{bmatrix} \in \mathbb{R}^{(n_c+1) \times n_c}$ má právě o jeden řádek víc než má sloupců. Tento jeden řádek lze zapsat jako lineární kombinaci zbylých n_c řádků s koeficienty

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_{n_c}]^T \neq \mathbf{0}$$

Potom platí:

$${}_z\bar{\mathbf{f}}^T = \boldsymbol{\lambda}^T \mathbf{g}_z \quad \text{nebo} \quad {}_z\bar{\mathbf{f}} = \mathbf{g}_z^T \boldsymbol{\lambda} \quad (2.95)$$

Dosazením (2.95) do (2.93) se získá:

$$\boldsymbol{\lambda}^T \mathbf{g}_z \delta\mathbf{z} = -{}_q\bar{\mathbf{f}}^T \delta\mathbf{q} \quad (2.96)$$

$$-\boldsymbol{\lambda}^T \mathbf{g}_q \delta\mathbf{q} = -{}_q\bar{\mathbf{f}}^T \delta\mathbf{q} \quad (2.97)$$

Transponování rovnice (2.97):

$${}_q\bar{\mathbf{f}} = \mathbf{g}_q^T \boldsymbol{\lambda} \quad (2.98)$$

A finální spojení rovnice (2.95) s (2.98):

$$\begin{bmatrix} {}_z\bar{\mathbf{f}} \\ {}_q\bar{\mathbf{f}} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_z^T \\ \mathbf{g}_q^T \end{bmatrix} \boldsymbol{\lambda} \quad (2.99)$$

Nebo kompaktnější zápis v maticové formě:

$${}^c\bar{\mathbf{f}} = \mathbf{g}_s^T \boldsymbol{\lambda} \quad (2.100)$$

Pohybové rovnice jsou psané v souřadnicích rychlosti \mathbf{v} místo derivace vektoru souřadnic $\dot{\mathbf{s}}$, a proto je nutné tento přepočít zahrnout při počítání reakčních sil. Protože tyto dva vektory se liší pouze v úhlových souřadnicích, stačí přepočítat právě

je. Z principu virtuálních prací musí platit, že virtuální výkon sil na souřadnicích $\dot{\mathbf{s}}$ musí být stejný jako na souřadnicích \mathbf{v} , tedy platí:

$$[M_{xi}, M_{yi}, M_{zi}] \underbrace{\begin{bmatrix} \delta \dot{\phi}_{xi} \\ \delta \dot{\phi}_{yi} \\ \delta \dot{\phi}_{zi} \end{bmatrix}}_{\delta \dot{\boldsymbol{\eta}}_i} = [{}^i M_{xi}, {}^i M_{yi}, {}^i M_{zi}] \underbrace{\begin{bmatrix} \delta^i \omega_{x1i} \\ \delta^i \omega_{y1i} \\ \delta^i \omega_{z1i} \end{bmatrix}}_{{}^i \delta \boldsymbol{\omega}_{i1}} \quad (2.101)$$

Při použití inverzního přepočtu definovaného v rovnici (2.72):

$$\delta {}^i \boldsymbol{\omega}_{i1} = \mathbf{S}_{i1}(\eta_i) \mathbf{H}_i^{-1}(\eta_i) \delta \dot{\boldsymbol{\eta}}_i \quad (2.102)$$

Vykrácení virtuálních rychlostí:

$$[M_{xi}, M_{yi}, M_{zi}] \cancel{\begin{bmatrix} \delta \dot{\phi}_{xi} \\ \delta \dot{\phi}_{yi} \\ \delta \dot{\phi}_{zi} \end{bmatrix}} = [{}^i M_{xi}, {}^i M_{yi}, {}^i M_{zi}] \mathbf{S}_{i1}(\eta_i) \mathbf{H}_i^{-1}(\eta_i) \cancel{\begin{bmatrix} \delta \dot{\phi}_{xi} \\ \delta \dot{\phi}_{yi} \\ \delta \dot{\phi}_{zi} \end{bmatrix}} \quad (2.103)$$

Transponování rovnice:

$$\begin{bmatrix} M_{xi} \\ M_{yi} \\ M_{zi} \end{bmatrix} = (\mathbf{S}_{i1}(\eta_i) \mathbf{H}_i^{-1}(\eta_i))^T \begin{bmatrix} {}^i M_{xi} \\ {}^i M_{yi} \\ {}^i M_{zi} \end{bmatrix} \quad (2.104)$$

A přidání sil k momentům:

$$\underbrace{\begin{bmatrix} {}^1 f_{xi} \\ {}^1 f_{yi} \\ {}^1 f_{zi} \\ M_{xi} \\ M_{yi} \\ M_{zi} \end{bmatrix}}_{= {}^c \bar{\mathbf{f}}_i} = \underbrace{\begin{bmatrix} \mathbf{E}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & (\mathbf{S}_{i1}(\eta_i) \mathbf{H}_i^{-1}(\eta_i))^T \end{bmatrix}}_{\mathbf{T}_i^{-T}(\eta)} \underbrace{\begin{bmatrix} {}^1 f_{xi} \\ {}^1 f_{yi} \\ {}^1 f_{zi} \\ {}^i M_{xi} \\ {}^i M_{yi} \\ {}^i M_{zi} \end{bmatrix}}_{= {}^c \mathbf{f}_i} \quad (2.105)$$

Porovnáním rovnic (2.100) a (2.105) zapsaných za použití známých symbolů vznikne:

$${}^c \bar{\mathbf{f}}_i = \mathbf{T}_i^{-T}(\boldsymbol{\eta}) {}^c \mathbf{f}_i = \mathbf{g}_{is}^T \boldsymbol{\lambda}_i \quad (2.106)$$

Finální úprava je přenásobením rovnice (2.106) výrazem \mathbf{T}^T z leva:

$${}^c \mathbf{f}_i = \mathbf{T}_i^T(\boldsymbol{\eta}) \mathbf{g}_{is}^T \boldsymbol{\lambda}_i \quad (2.107)$$

A tím vznikne vztah pro výpočet reakčních sil jednoho tělesa.

2.5 Pohybové rovnice tělesa s vazbami

Kompletní sada rovnic dynamiky jednoho tělesa se skládá z Newton-Eulerových rovnic tělesa bez vazeb (odvozených v sekci 2.2) doplněných o reakční síly ${}^c\mathbf{f}$ (odvozených v sekci 2.4):

$$\mathbf{M}(\mathbf{s})\dot{\mathbf{v}} = \mathbf{f} + {}^c\mathbf{f} + \mathbf{q}_G \quad (2.108)$$

$${}^c\mathbf{f} = \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s})\lambda \quad (2.109)$$

Dále z n_c vazbových rovnic:

$$\mathbf{g}_s(\mathbf{s}) = 0 \quad (2.110)$$

A z diferenciálních rovnic kinematiky odvozených v sekci 2.3:

$$\dot{\mathbf{s}} = \mathbf{T}(\mathbf{s})\mathbf{v} \quad (2.111)$$

$$\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\dot{\mathbf{v}} = \boldsymbol{\beta}_c(\mathbf{s}, \dot{\mathbf{s}}) \quad (2.112)$$

Rovnice lze přepsat do známého tvaru z Lagrangeových rovnic smíšeného typu:

$$\begin{bmatrix} \mathbf{M}(\mathbf{s}) & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \\ \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s}) & \mathbf{0}_{n_c \times n_c} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f} + \mathbf{q}_G \\ \boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v}) \end{bmatrix} \quad (2.113)$$

Případně je možné maticovou rovnici rozšířit i o rovnici (2.111):

$$\begin{bmatrix} \mathbf{E}_{6 \times 6} & \mathbf{0}_{6 \times 6} & \mathbf{0}_{6 \times n_c} \\ \mathbf{0}_{6 \times 6} & \mathbf{M}(\mathbf{s}) & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \\ \mathbf{0}_{n_c \times 6} & \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s}) & \mathbf{0}_{n_c \times n_c} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{v}} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{T}(\mathbf{s})\mathbf{v} \\ \mathbf{f} + \mathbf{q}_G \\ \boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v}) \end{bmatrix} \quad (2.114)$$

- $\mathbf{M}(\mathbf{s})$ je matice hmotnosti tělesa definovaná v (2.57)
- $\mathbf{T}(\mathbf{s})$ je matice přepočtu vektoru rychlostí \mathbf{v} na derivaci vektoru souřadnic $\dot{\mathbf{s}}$ definová v (2.74)
- \mathbf{f} je vektor vnějších sil působící na těleso definovaný v (2.59)
- \mathbf{q}_G je vektor gyroskopických sil definovaný v (2.60)
- $\dot{\mathbf{s}}$ je derivace vektoru souřadnic jednoho tělesa (2.63)
- \mathbf{v} je vektor rychlostí definovaný v (2.64)
- $\dot{\mathbf{v}}$ je derivace vektoru rychlostí

- $\beta_c(\mathbf{s}, \mathbf{v})$ je vektor pravé strany rovnic kinematiky
- $\mathbf{g}_s(\mathbf{s})$ je Jacobiho matice vazeb
- n_c je počet vazbových rovnic
- $\boldsymbol{\lambda}$ je vektor Lagrangeových multiplikátorů

2.6 Pohybové rovnice pro soustavu tuhých těles

Pro vytvoření soustavy rovnic pro více těles je potřeba navrhnout strukturu vektorů popisující jejich dynamiku. V předchozí sekci (sekce 2.5) byly představeny rovnice popisující dynamiku jednoho tělesa s vazbami. V této sekci bude představen způsob, jak spojit dynamiku více těles v soustavu lineárních rovnic. V následujících rovnicích n_B značí počet těles.

Pro popis dynamiky více těles bude použito následujících vektorů. Vektor souřadnic pro popis aktuální konfigurace \mathbf{s} již byl přestaven v sekci 2.3 stejně jako vektor rychlostí \mathbf{v} .

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_n \end{bmatrix} \in \mathbb{R}^{6n_B} \quad \mathbf{s}_i = \begin{bmatrix} \mathbf{r}_i \\ \boldsymbol{\eta}_i \end{bmatrix} \in \mathbb{R}^6 \quad \mathbf{r}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \in \mathbb{R}^3 \quad \boldsymbol{\eta}_i = \begin{bmatrix} \phi_{xi} \\ \phi_{yi} \\ \phi_{zi} \end{bmatrix} \in \mathbb{R}^3 \quad (2.115)$$

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} \in \mathbb{R}^{6n_B} \quad \mathbf{v}_i = \begin{bmatrix} \dot{\mathbf{r}}_i \\ {}^i\boldsymbol{\omega}_{1i} \end{bmatrix} \in \mathbb{R}^6 \quad \dot{\mathbf{r}}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} \in \mathbb{R}^3 \quad {}^i\boldsymbol{\omega}_{1i} = \begin{bmatrix} {}^i\omega_{xi} \\ {}^i\omega_{yi} \\ {}^i\omega_{zi} \end{bmatrix} \in \mathbb{R}^3 \quad (2.116)$$

První potřebnou maticí je matice přepočtu vektoru rychlostí na vektor derivace souřadnic $\mathbf{T}(\mathbf{s})$. Tato blokově diagonální matice je složena z dílčích matic $\mathbf{T}_i(\mathbf{s})$ definovaných v (2.73):

$$\mathbf{T}(\mathbf{s}) = \text{diag}(\mathbf{T}_1(\mathbf{s}_1), \mathbf{T}_1(\mathbf{s}_2), \dots, \mathbf{T}_{n_B}(\mathbf{s}_{n_B})) \quad (2.117)$$

Matice hmotnosti všech těles $\mathbf{M}(\mathbf{s})$ je blokově diagonální matice složená z dílčích matic hmotnosti z rovnice (2.57):

$$\mathbf{M} = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{n_B}) \quad (2.118)$$

Vektor vnějších zobecněných sil působící na tělesa je složen z dílčích vektorů zobecněných sil \mathbf{f}_i definovaných v rovnici (2.59):

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{n_B} \end{bmatrix} \quad (2.119)$$

Vektor reakčních sil ${}^c\mathbf{f}$ pro dodržení vazbových podmínek je složen obdobně jako vektor vnějších sil. Namísto vnějších sil obsahuje reakční síly působící na jednotlivá tělesa. Lze jej vypočítat použitím matice $\mathbf{T}(\mathbf{s})$, Jacobiho matice vazeb $\mathbf{g}_s(\mathbf{s})$ a vektoru Lagrangeových multiplikátorů $\boldsymbol{\lambda}$:

$${}^c\mathbf{f} = \begin{bmatrix} {}^c\mathbf{f}_1 \\ {}^c\mathbf{f}_2 \\ \vdots \\ {}^c\mathbf{f}_{n_c} \end{bmatrix} = \mathbf{T}^T \mathbf{g}_s^T \boldsymbol{\lambda} \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{n_c} \end{bmatrix} \in \mathbb{R}^{n_c} \quad \mathbf{g}_s(\mathbf{s}) \in \mathbb{R}^{n_c, 6n_B} \quad (2.120)$$

Kde $\mathbf{g}(\mathbf{s})$ je vektor n_c vazbových rovnic, u kterých se předpokládá, že jsou na sobě nezávislé, resp. že platí $\text{rank}(\mathbf{g}_s(\mathbf{s})) = n_c$:

$$\mathbf{g}(\mathbf{s}) = \begin{bmatrix} g_1(\mathbf{s}) \\ g_2(\mathbf{s}) \\ \vdots \\ g_{n_c}(\mathbf{s}) \end{bmatrix} = \mathbf{0}_{n_c \times 1} \in \mathbb{R}^{n_c} \quad (2.121)$$

Posledním potřebným vektorem je vektor pravé strany kinematických rovnic $\boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v})$.

$$\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\dot{\mathbf{v}} = \boldsymbol{\beta}_c \in \mathbb{R}^{n_c} \quad \boldsymbol{\beta}_c = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n_c} \end{bmatrix} \quad (2.122)$$

Následně je možné vytvořit rovnici popisující dynamiku soustavy tuhých těles v obdobném tvaru, jako tomu bylo pro jedno těleso s vazbami. Chytrou volbou struktury vektorů se podařilo vytvořit popis umožňující přidání libovolnému počtu těles.

$$\begin{bmatrix} \mathbf{E}_{6n_b \times 6n_b} & \mathbf{0}_{6n_b \times 6n_b} & \mathbf{0}_{6n_b \times n_c} \\ \mathbf{0}_{6n_b \times 6n_b} & \mathbf{M}(\mathbf{s}) & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \\ \mathbf{0}_{n_c \times 6n_b} & \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s}) & \mathbf{0}_{n_c \times n_c} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{T}(\mathbf{s})\mathbf{v} \\ \mathbf{f} + \mathbf{q}_G \\ \boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v}) \end{bmatrix} \quad (2.123)$$

2.7 Porušení vazbových rovnic

Problémem při řešení soustavy DAE rovnic je splnění vazbových rovnic. V pohybových rovnicích (2.123) jsou vazbové rovnice obsaženy pouze ve formě jejich druhé derivace (viz sekce 2.3), nikoliv jejich první derivace nebo nederivovaná podoba. Při numerické integraci tato chyba roste a vazbové rovnice přestávají být splněny.

Pro řešení tohoto problému jsou k dispozici dva typy nástrojů. Prvním typem je převod DAE rovnic do ODE, použitím převodu fyzikálních souřadnic na nezávislé. To je možné provést např. pomocí projekční matice. Druhý způsob nepřevádí problém do nezávislých souřadnic, ale využívá Baumgartovy stabilizace označované po jejím autorovi [25, s. 361]. Tě bude využito ve vytvářeném softwaru.

Stabilizace využívá faktu, že vazbové rovnice i její derivace jsou rovny nule:

$$\mathbf{g}(\mathbf{s}) = \mathbf{0}, \quad \dot{\mathbf{g}}(\mathbf{s}) = \mathbf{0}, \quad \ddot{\mathbf{g}}(\mathbf{s}) = \mathbf{0} \quad (2.124)$$

Sestavením diferenciální rovnice druhého řádu s koeficienty $\alpha, \beta \in \mathbb{R}$:

$$\ddot{\mathbf{g}}(\mathbf{s}) + 2\alpha\dot{\mathbf{g}}(\mathbf{s}) + \beta^2\mathbf{g}(\mathbf{s}) = \mathbf{0} \quad (2.125)$$

Lze ukázat, že při volbě koeficientů $\alpha = \beta = 1$, je její řešení ve formě exponenciálního útlumu:

$$\mathbf{g}(\mathbf{s}) = e^{-t}(\mathbf{C}_1 + \mathbf{C}_2 t) \quad (2.126)$$

Dosazením za derivace vazbových rovnic z (2.76) a (2.80):

$$\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\dot{\mathbf{v}} - \beta_c(\mathbf{s}, \dot{\mathbf{s}}) + 2\alpha\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\mathbf{v} + \beta^2\mathbf{g}(\mathbf{s}) = \mathbf{0} \quad (2.127)$$

Rovnice dynamiky všech těles (2.123) lze tedy podle rovnice (2.127) doplnit o členy stabilizace:

$$\begin{bmatrix} \mathbf{E}_{6n_b \times 6n_b} & \mathbf{0}_{6n_b \times 6n_b} & \mathbf{0}_{6n_b \times n_c} \\ \mathbf{0}_{6n_b \times 6n_b} & \mathbf{M}(\mathbf{s}) & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \\ \mathbf{0}_{n_c \times 6n_b} & \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s}) & \mathbf{0}_{n_c \times n_c} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{v}} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{T}(\mathbf{s})\mathbf{v} \\ \mathbf{f} + \mathbf{q}_G \\ \beta_c(\mathbf{s}, \mathbf{v}) - 2\alpha\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\mathbf{v} - \beta^2\mathbf{g}(\mathbf{s}) \end{bmatrix} \quad (2.128)$$

2.8 Silové účinky od pružin, tlumičů a lan

Jedním z požadavků na výsledný software je, aby umožnil separaci silových účinků jednotlivých lan, tlumičů a pružin na matici distribuce sil a skalární velikost sil.

Vektor vnějších sil působící na tělesa (2.119) lze rozdělit na síly pružin \mathbf{f}_s , tlumičů \mathbf{f}_d a lan \mathbf{f}_c , případně na další síly $f_{další}$, které v této části nejsou uvažovány:

$$\mathbf{f} = \mathbf{f}_s + \mathbf{f}_d + \mathbf{f}_c + (f_{další}) \quad (2.129)$$

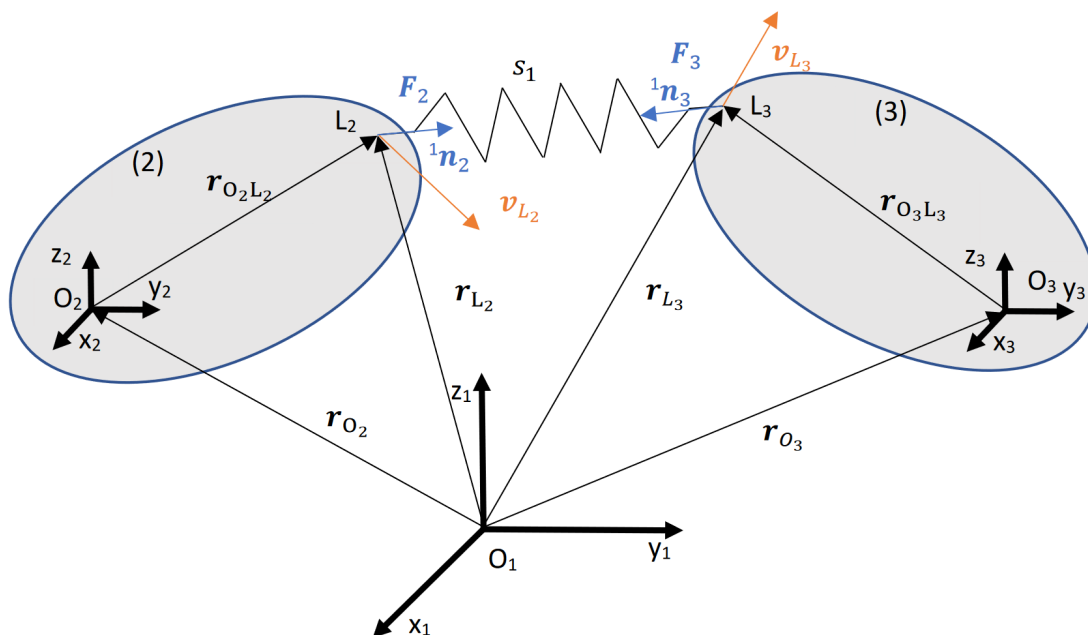
Separaci na matici distribuce a skalární velikost síly je myšleno následující rozdělení:

$$\mathbf{f} = \mathbf{T}_s \mathbf{F}_s + \mathbf{T}_d \mathbf{F}_d + \mathbf{T}_c \mathbf{F}_c \quad (2.130)$$

Kde \mathbf{T}_s , \mathbf{T}_d a \mathbf{T}_c jsou matice distribuce sil pružin, tlumičů a lan. \mathbf{F}_s , \mathbf{F}_d a \mathbf{F}_c jsou vektory velikostí sil v pružinách, tlumičích a lanech.

2.8.1 Výpočet distribuční matice pro pružiny a tlumiče

V první řadě bude odvozen výpočet distribuční matice pro pružiny a tlumiče. Předpokládá se, že všechny pružiny i tlumiče vždy spojují dva body na tělesech. Při odvození se vychází z obr. 2.6, na kterém jsou vidět dvě tělesa spojená pružinou. Pružina je spojena s body L_2 na tělese 2 a L_3 na tělese 3.



Obr. 2.6: Pružina mezi dvěma tělesy

Cílem je najít vektor popisující silové účinky pružiny na soustavu těles \mathbf{f}_s . Pro jeho složky platí:

$$\mathbf{f}_s = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix} \quad \mathbf{f}_2 = \begin{bmatrix} {}^1\mathbf{F}_2 \\ {}^2\mathbf{M}_2 \end{bmatrix} \quad \mathbf{f}_3 = \begin{bmatrix} {}^1\mathbf{F}_3 \\ {}^3\mathbf{M}_3 \end{bmatrix} \quad (2.131)$$

Vektor \mathbf{f}_s se skládá z vektorů ${}^1\mathbf{F}_2$ a ${}^2\mathbf{M}_2$ působících na těleso označené číslem 2 a z vektorů ${}^1\mathbf{F}_3$ a ${}^3\mathbf{M}_3$ působících na těleso s číslem 3. Dalším krokem pro získání dis-

tribuční matice je vyjádření vektoru síly ${}^1\mathbf{F}_2$ za pomoci normály ve směru působení a absolutní velikosti jeho silového účinku. Pro sílu ${}^1\mathbf{F}_2$ platí:

$${}^1\mathbf{F}_2 = {}^1\mathbf{n}_2 \| \mathbf{F}_2 \| \quad (2.132)$$

Kde ${}^1\mathbf{n}_2$ je jednotkový vektor ve směru síly ${}^1\mathbf{F}_2$. Tímto způsobem lze problém rozdělit na hledání jednotkového vektoru mezi tělesy 2 a 3 a velikosti síly v pružině. Pro nalezení jednotkového vektoru lze použít normovaný rozdíl mezi souřadnicemi uzlů L_2 a L_3 , mezi kterými je pružina uchycena. Jednotkový vektor ${}^1\mathbf{n}_2$ lze vypočítat jako:

$${}^1\mathbf{n}_2 = \frac{{}^1\mathbf{r}_{L_3} - {}^1\mathbf{r}_{L_2}}{\| {}^1\mathbf{r}_{L_3} - {}^1\mathbf{r}_{L_2} \|} \quad (2.133)$$

Druhým problémem je výpočet momentu ${}^2\mathbf{M}_2$. To je moment vyvolaný působením síly na těleso. Moment síly je třeba vyjadřovat v souřadnicovém systému tělesa podle (2.59) tzn. v souřadnicovém systému 2. Lze jej vypočítat jako:

$${}^2\mathbf{M}_2 = {}^2\mathbf{r}_{O_2L_2} \times {}^2\mathbf{F}_2 = {}^2\tilde{\mathbf{r}}_{O_2L_2} {}^2\mathbf{F}_2 \quad (2.134)$$

Stejně jako v předchozím kroku se provede separace velikosti síly od jejího směru za použití jednotkové normály. V předchozím kroku byla síla rozložena při vyjádření v souřadnicovém systému rámu. Z tohoto důvodu je i zde síla transformována do souřadnicového systému rámu, a pak je aplikován stejný rozklad.

$${}^2\mathbf{M}_2 = {}^2\tilde{\mathbf{r}}_{O_2L_2} {}^2\mathbf{F}_2 = {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{F}_2 = {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_2 \| \mathbf{F}_2 \| \quad (2.135)$$

Vektor \mathbf{f}_2 lze pak zapsat pomocí rovnic (2.132) a (2.135) ve tvaru:

$$\mathbf{f}_2 = \begin{bmatrix} {}^1\mathbf{F}_2 \\ {}^2\mathbf{M}_2 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{n}_2 \\ {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_2 \end{bmatrix} \| \mathbf{F}_2 \| \quad (2.136)$$

Obdobným způsobem lze postupovat i v případě vektoru \mathbf{f}_3 :

$$\mathbf{f}_3 = \begin{bmatrix} {}^1\mathbf{F}_3 \\ {}^3\mathbf{M}_3 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{n}_3 \\ {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_3 \end{bmatrix} \| \mathbf{F}_3 \| \quad (2.137)$$

Obě síly (\mathbf{F}_2 , \mathbf{F}_3) musí být stejně velké, protože jsou vyvolány stejnou pružinou a platí zákon akce a reakce:

$$\| \mathbf{F}_2 \| = \| \mathbf{F}_3 \| = \| \mathbf{F}_{s_1} \| = F_{s_1} \quad (2.138)$$

Kde $\| \mathbf{F}_{s_1} \|$ je velikost síly v laně s_1

Vložení rovnic (2.136), (2.137) a (2.138) do rovnice (2.131):

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{n}_2 \\ {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_2 \\ {}^1\mathbf{n}_3 \\ {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_3 \end{bmatrix} \|\mathbf{F}_{s_1}\| = \mathbf{T}_s \mathbf{F}_s \quad (2.139)$$

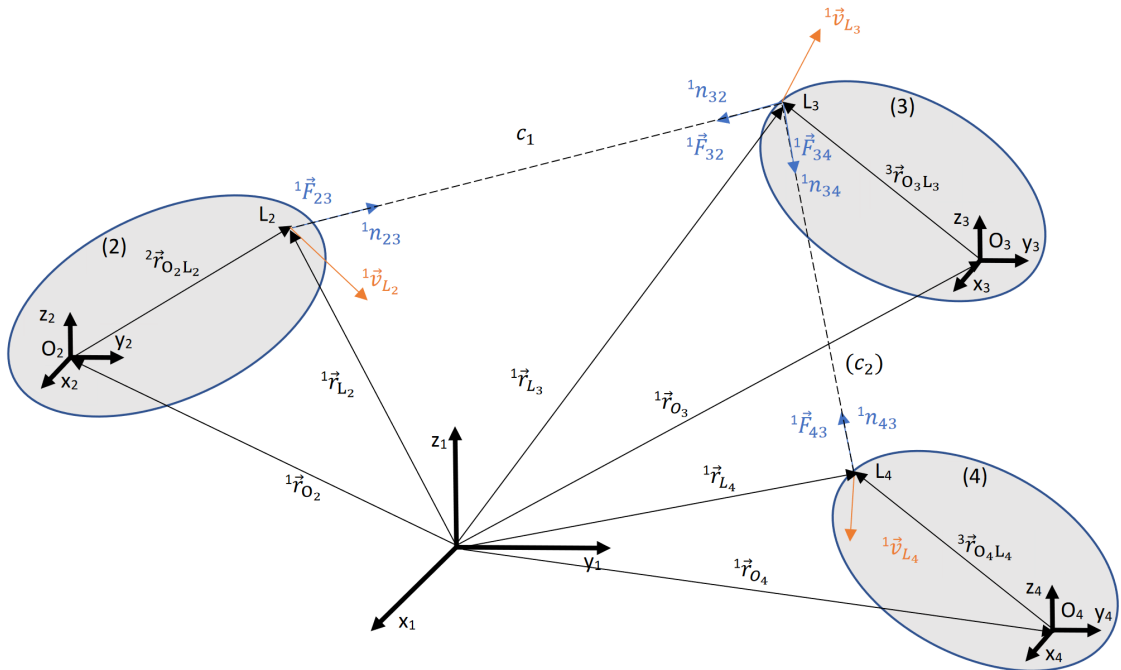
Zároveň platí, že normály ${}^1\mathbf{n}_2$ a ${}^1\mathbf{n}_3$ jsou shodné až na orientaci. To znamená ${}^1\mathbf{n}_2 = -{}^1\mathbf{n}_3$, rovnici (2.139) lze psát jako:

$$\mathbf{f} = \begin{bmatrix} {}^1\mathbf{n}_2 \\ {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_2 \\ -{}^1\mathbf{n}_2 \\ -{}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_2 \end{bmatrix} \|\mathbf{F}_{s_1}\| = \mathbf{T}_s \mathbf{F}_s \quad (2.140)$$

Problém se tedy tímto rozpadl na problém určení matice \mathbf{T}_c , kterou lze vypočítat z aktuální konfigurace těles a vektoru sil v pružinách \mathbf{F}_s , jehož výpočet bude popsán dále subsekcí 2.8.3.

2.8.2 Výpočet distribuční matice pro lano spojující více uzlů

V přechodzí subsekcí byl popsán způsob odvození distribuční matice \mathbf{T}_s pro jednu pružinu nebo tlumič. Zde bude odvozena distribuční matice \mathbf{T}_c pro lana. Velkou změnou oproti pružinám, pro které byla odvozena distribuční matice v předchozí sekci je, že lana mohou propojovat více těles najednou. Např. na obr. 2.7 jsou jedním lanem (čárkovanou čarou znázorněným) propojena tři tělesa.



Obr. 2.7: Lano mezi třemi tělesy

Na první pohled je zřejmé, že pokud by bylo lano propojeno pouze mezi dvěma

uzly, celý problém je identický s problémem získání distribuční matice \mathbf{T}_s . Pro odvození distribuční matice pro lana spojující více těles je využito základního předpokladu, že je síla v každém místě lana stejná. Model provlečeného lana tedy nebere v potaz tření v průvlacích jednotlivých těles. Díky tomu je možné rozdělit problém na dílčí problémy.

Předpokládejme nejprve, že lano je složeno z dílčích lan vždy spojující jen 2 uzly. První dílčí lano, na obrázku označené jako c_1 , spojuje body L_2 a L_3 . Druhé, označené jako (c_2), spojuje body L_3 a L_4 . Distribuční matice pro tato dílčí lana jsou shodné s distribuční maticí odvozenou pro pružinu (2.139). Celková distribuční matice zahrnující obě dílčí lana je zobrazena v rovnici (2.141)

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{n}_{23} & \mathbf{0}_3 \\ {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_{23} & \mathbf{0}_3 \\ {}^1\mathbf{n}_{32} & {}^1\mathbf{n}_{34} \\ {}^2\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{32} & {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{34} \\ \mathbf{0}_3 & {}^1\mathbf{n}_{43} \\ \mathbf{0}_3 & {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{34} \end{bmatrix} \begin{bmatrix} \|\mathbf{F}_{c_1}\| \\ \|\mathbf{F}_{c_2}\| \end{bmatrix} \quad (2.141)$$

Aby se ze dvou dílčích lan stalo lano provlečené, je třeba přidat předpoklad stejné síly v obou lanech. Matematicky je to zapsáno jako $\|\mathbf{F}_{c_1}\| = \|\mathbf{F}_{c_2}\|$. Díky tomu je možné sloupce distribuční matice sečíst a získat tak celkovou distribuční matici silových účinků od provlečeného lana.

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{bmatrix} = \begin{bmatrix} {}^1\mathbf{n}_{23} \\ {}^2\tilde{\mathbf{r}}_{O_2L_2} \mathbf{S}_{12}^T {}^1\mathbf{n}_{23} \\ {}^1\mathbf{n}_{32} + {}^1\mathbf{n}_{34} \\ {}^2\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{32} + {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{34} \\ {}^1\mathbf{n}_{43} \\ {}^3\tilde{\mathbf{r}}_{O_3L_3} \mathbf{S}_{13}^T {}^1\mathbf{n}_{34} \end{bmatrix} \|\mathbf{F}_{c_1}\| \quad (2.142)$$

Obdobným způsobem by se postupovalo i v případech, kdy by provlečené lano spojovalo více než tři uzly.

2.8.3 Výpočet velikosti sil v pružinách

Pružiny jsou modelovány jako lineární. Velikost síly v pružině se vypočítá jako:

$$F_i = K_i(l_i - l_{0i}) \quad (2.143)$$

- K_i [N/m] je tuhost pružiny
- l_i [m] je aktuální délka pružiny
- l_{0i} [m] je volná délka pružiny (délka nezatížené pružiny)

Aktuální délku pružiny l_i je možné vypočítat pomocí normy jako:

$$l_i = \|\mathbf{r}_k - \mathbf{r}_j\| \quad (2.144)$$

Kde \mathbf{r}_k a \mathbf{r}_j jsou aktuální rádiusvektory uzlů, které pružina spojuje. Tyto vektory lze vypočítat za použití aktuální konfigurace soustavy těles a lokálního rádiusvektoru jako:

$$l_i = \|\mathbf{r}_k - \mathbf{r}_j\| = \|\mathbf{r}_{O_k} + \mathbf{S}_{1k}^k \mathbf{r}_{O_k L} - \mathbf{r}_{O_j} - \mathbf{S}_{1j}^j \mathbf{r}_{O_j L}\| \quad (2.145)$$

2.8.4 Výpočet velikosti sil v tlumičích

Tlumiče jsou modelovány jako lineární. Pro výpočet tlumící síly se vychází z obr. 2.6. Stejně jako v případě pružiny se předpokládá, že kladná velikost tlumící síly táhne uzly k sobě. Pro tlumič umístěný mezi uzly L_2 a L_3 se tlumící síla F_d vypočítá jako:

$$F_d = B \mathbf{n}_2^T (\mathbf{v}_{L_3} - \mathbf{v}_{L_2}) \quad (2.146)$$

- B [Ns/m] je koeficient tlumení tlumiče
- \mathbf{n}_2 [/] je jednotková normála mezi uzlem L_2 a L_3
- \mathbf{v}_{L_2} a \mathbf{v}_{L_3} [m/s] jsou vektory rychlosti uzlu L_2 a L_3 v souřadnicovém systému rámu

Rychlost uzlů lze vypočítat časovou derivací polohového vektoru uzlu jako:

$$\mathbf{v}_{L_2} = \dot{\mathbf{r}}_{L_2} = \dot{\mathbf{r}}_{O_2} + \mathbf{S}_{12}^2 \boldsymbol{\Omega}_{12}^2 \mathbf{r}_{O_2 L_2} \quad (2.147)$$

2.8.5 Výpočet velikosti sil v lanech

Výpočet sil v lanech je složitější než výpočet sil v pružinách a tlumičích. Model lana je složen z paralelní kombinace pružiny a tlumiče. Konstanty tuhosti jsou proměnné v závislosti na aktuální volné délce lana (nepřímá úměra). Výpočet síly v laně je zde rozdělen na dvě části. První část síly v laně je podobná výpočtu síly v pružině. Hlavním rozdílem je, že lano může být provlečeno přes více uzlů a je nutné posčítat všechny části lana. Síla od tuhostní složky lana i F_{ck} se vypočítá jako:

$$F_{ck} = \frac{k_i}{l_{0i}} \left(\sum_j (l_{ij}) - l_{0i} \right) \quad (2.148)$$

- k_i [$\frac{mN}{m}$] je měrná tuhost lana
- l_{0i} [m] je volná délka lana

- $l_{ij} [m]$ je aktuální délka úseku j lana i

Suma přes délky všech úseků lana odpovídá celkové aktuální délce lana.

Druhou složkou síly v laně je složka tlumení. Ta vychází z časové derivace aktuální délky lana. Posčítáním všech časových derivací délek jednotlivých úseků je možné vypočítat tlumící složku síly v laně jako:

$$F_{cd} = \frac{b_i}{l_{0i}} \left(\sum_j ({}^1\mathbf{n}_j^T ({}^1\mathbf{v}_{j+1} - {}^1\mathbf{v}_j)) \right) \quad (2.149)$$

- $b_i [\frac{mNs}{m}]$ je měrné tlumení lana
- $l_{0i} [m]$ je volná délka lana
- ${}^1\mathbf{n}_j [/]$ je jednotková normála mezi uzly
- $\mathbf{v}_j [\frac{m}{s}]$ je vektor rychlosti uzlu j

Celková síla v laně je následně dána jako:

$$F_c = F_{ck} + F_{cd} \quad (2.150)$$

U lan je potřeba kontrolovat, jestli nedošlo:

- Ke ztrátě napnutí lana (tzn. $F_{ck} < 0$)
- K přenosu tlakové síly (tzn. $F_c < 0$)

Musí být splněny obě podmínky současně. Prověšené lano nemůže přenášet žádnou sílu. Stejně tak lano nemůže přenášet tlakovou sílu. Pokud je alespoň jedna z podmínek splněna, je síla v laně nulová. Tyto dvě podmínky vnáší do modelu silnou nelinearitu.

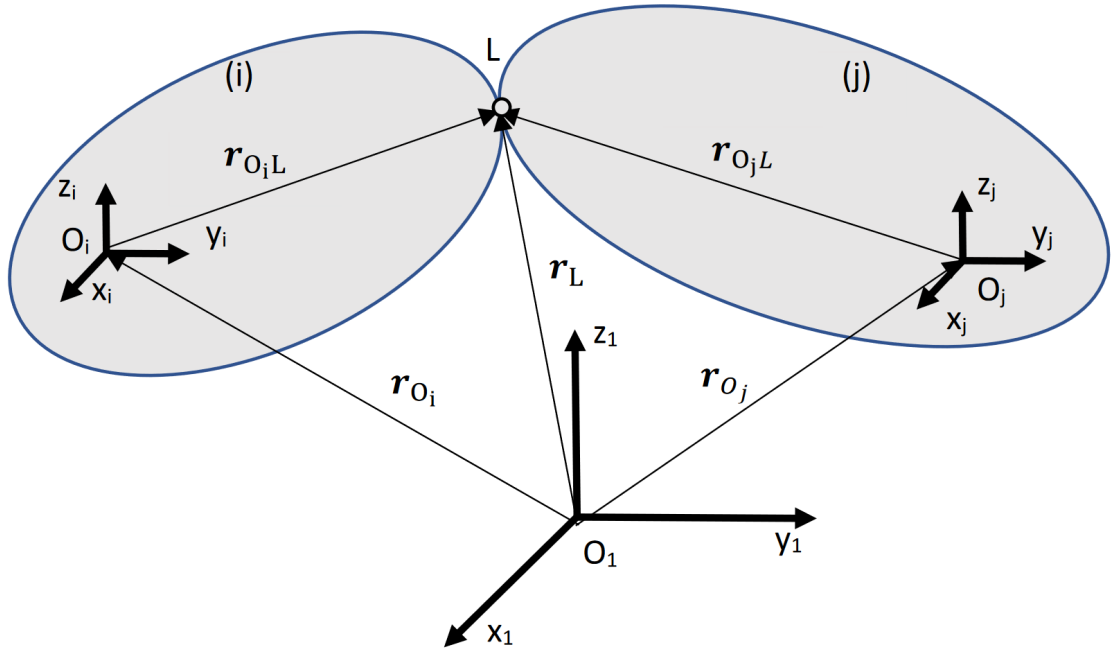
2.9 Odvození Jacobiho matice pro sférickou vazbu

V této sekci bude odvozena Jacobiho matice $\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})$ a vektor pravé strany $\beta_c(\mathbf{s}, \mathbf{v})$ kinematických rovnic pro sférickou vazbu spojující dvě tělesa. Dvě tělesa na obr. 2.8 jsou spojena sférickou vazbou v bodě L. Tělesa jsou označena jako těleso i a těleso j . Při odvození se předpokládá, že rádiusvektor \mathbf{r}_L lze vyjádřit identicky za pomoci lokálního souřadnicového systému tělesa i tak j .

$${}^1\mathbf{r}_L = {}^1\mathbf{r}_L \quad (2.151)$$

Vyjádřením \mathbf{r}_L za pomoci souřadnicových systémů tělesa i a j :

$${}^1\mathbf{r}_{O_i} + {}^1\mathbf{r}_{O_iL} = {}^1\mathbf{r}_{O_j} + {}^1\mathbf{r}_{O_jL} \quad (2.152)$$



Obr. 2.8: Spojení dvou těles sférickou vazbou

Přechod lokálních rádiusvektorů do bází souřadnicových systémů těles za pomoci matice transformace \mathbf{S}_{12} :

$${}^1\mathbf{r}_{O_i} + \mathbf{S}_{1i} {}^i\mathbf{r}_{O_iL} = {}^1\mathbf{r}_{O_j} + \mathbf{S}_{1j} {}^j\mathbf{r}_{O_jL} \quad (2.153)$$

Následuje úprava rovnice (2.153) tak, aby vznikla vektorová rovnice vazeb $\mathbf{g}(\mathbf{s})$:

$$\mathbf{g}(\mathbf{s}) = {}^1\mathbf{r}_{O_i} + \mathbf{S}_{1i} {}^i\mathbf{r}_{O_iL} - {}^1\mathbf{r}_{O_j} - \mathbf{S}_{1j} {}^j\mathbf{r}_{O_jL} = \mathbf{0}_{3 \times 1} \quad (2.154)$$

Časovou derivací rovnice (2.154) se získá:

$$\begin{aligned} \dot{\mathbf{g}}(\mathbf{s}) &= \mathbf{g}_s(\mathbf{s})\dot{\mathbf{s}} = \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\mathbf{v} = \\ &{}^1\dot{\mathbf{r}}_{O_i} + \mathbf{S}_{1i} {}^i\tilde{\boldsymbol{\Omega}}_{1i} {}^i\mathbf{r}_{O_iL} - {}^1\dot{\mathbf{r}}_{O_j} - \mathbf{S}_{1j} {}^j\tilde{\boldsymbol{\Omega}}_{1j} {}^j\mathbf{r}_{O_jL} = \mathbf{0}_{3 \times 1} \end{aligned} \quad (2.155)$$

Použitím vlastnosti antisymetrických matic ${}^j\tilde{\boldsymbol{\omega}} {}^j\mathbf{r} = -{}^j\tilde{\mathbf{r}} {}^j\boldsymbol{\omega}$ přejde rovnice (2.155) do podoby:

$$\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\mathbf{v} = {}^1\dot{\mathbf{r}}_{O_i} - \mathbf{S}_{1i} {}^i\tilde{\mathbf{r}}_{O_iL} {}^i\boldsymbol{\omega}_{1i} - {}^1\dot{\mathbf{r}}_{O_j} + \mathbf{S}_{1j} {}^j\tilde{\mathbf{r}}_{O_jL} {}^j\boldsymbol{\omega}_{1j} = \mathbf{0}_{3 \times 1} \quad (2.156)$$

Převod do maticové podoby rovnice (2.156) bude vypadat:

$$\left[\mathbf{E}_{3 \times 3}, \quad -\mathbf{S}_{1i} \tilde{\mathbf{r}}_{O_i L} \quad -\mathbf{E}_{3 \times 3}, \quad \mathbf{S}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} \right] \begin{bmatrix} {}^1 \dot{\mathbf{r}}_{O_i} \\ {}^i \boldsymbol{\omega}_{1i} \\ {}^1 \dot{\mathbf{r}}_{O_j} \\ {}^j \boldsymbol{\omega}_{1j} \end{bmatrix} = \mathbf{0}_{3 \times 1} \quad (2.157)$$

Druhou časovou derivací vazbových rovnic (2.154) se získá:

$$\begin{aligned} \ddot{\mathbf{g}}(\mathbf{s}) = & {}^1 \ddot{\mathbf{r}}_{O_i} + \mathbf{S}_{1i} {}^i \boldsymbol{\Omega}_{1i} {}^i \boldsymbol{\Omega}_{1i} {}^i \mathbf{r}_{O_i L} + \mathbf{S}_{1i} {}^i \dot{\boldsymbol{\Omega}}_{1i} {}^i \mathbf{r}_{O_i L} \\ & - {}^1 \ddot{\mathbf{r}}_{O_j} - \mathbf{S}_{1j} {}^j \boldsymbol{\Omega}_{1j} {}^j \boldsymbol{\Omega}_{1j} {}^j \mathbf{r}_{O_j L} - \mathbf{S}_{1j} {}^j \dot{\boldsymbol{\Omega}}_{1j} {}^j \mathbf{r}_{O_j L} = \mathbf{0}_{3 \times 1} \end{aligned} \quad (2.158)$$

Aplikace vlastnosti antisymetrických matic ${}^j \tilde{\boldsymbol{\omega}} {}^j \mathbf{r} = -{}^j \tilde{\mathbf{r}} {}^j \boldsymbol{\omega}$:

$$\begin{aligned} & {}^1 \ddot{\mathbf{r}}_{O_i} - \mathbf{S}_{1i} {}^i \boldsymbol{\Omega}_{1i} {}^i \tilde{\mathbf{r}}_{O_i L} {}^i \boldsymbol{\omega}_{1i} - \mathbf{S}_{1i} {}^i \tilde{\mathbf{r}}_{O_i L} {}^i \dot{\boldsymbol{\omega}}_{1i} \\ & - {}^1 \ddot{\mathbf{r}}_{O_j} + \mathbf{S}_{1j} {}^j \boldsymbol{\Omega}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} {}^j \boldsymbol{\omega}_{1j} + \mathbf{S}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} {}^j \dot{\boldsymbol{\omega}}_{1j} = \mathbf{0}_{3 \times 1} \end{aligned} \quad (2.159)$$

Separací členů s druhou časovou derivací polohového vektoru a s první časovou derivací úhlové rychlosti těles vznikne:

$$\begin{aligned} & {}^1 \ddot{\mathbf{r}}_{O_i} - \mathbf{S}_{1i} {}^i \tilde{\mathbf{r}}_{O_i L} {}^i \dot{\boldsymbol{\omega}}_{1i} - {}^1 \ddot{\mathbf{r}}_{O_j} + \mathbf{S}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} {}^j \dot{\boldsymbol{\omega}}_{1j} \\ & = \mathbf{S}_{1i} {}^i \boldsymbol{\Omega}_{1i} {}^i \tilde{\mathbf{r}}_{O_i L} {}^i \boldsymbol{\omega}_{1i} - \mathbf{S}_{1j} {}^j \boldsymbol{\Omega}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} {}^j \boldsymbol{\omega}_{1j} \end{aligned} \quad (2.160)$$

A posledním krokem je převod do maticového tvaru:

$$\underbrace{\left[\mathbf{E}_{3 \times 3}, \quad -\mathbf{S}_{1i} {}^i \tilde{\mathbf{r}}_{L_i} \quad -\mathbf{E}_{3 \times 3}, \quad \mathbf{S}_{1j} {}^j \tilde{\mathbf{r}}_{L_j} \right]}_{\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\mathbf{v}} \underbrace{\begin{bmatrix} {}^1 \ddot{\mathbf{r}}_{O_i} \\ {}^i \dot{\boldsymbol{\omega}}_{1i} \\ {}^1 \ddot{\mathbf{r}}_{O_j} \\ {}^j \dot{\boldsymbol{\omega}}_{1j} \end{bmatrix}}_{\dot{\mathbf{v}}} = \underbrace{\mathbf{S}_{1i} {}^i \boldsymbol{\Omega}_{1i} {}^i \tilde{\mathbf{r}}_{O_i L} {}^i \boldsymbol{\omega}_{1i} - \mathbf{S}_{1j} {}^j \boldsymbol{\Omega}_{1j} {}^j \tilde{\mathbf{r}}_{O_j L} {}^j \boldsymbol{\omega}_{1j}}_{=\beta_c(\mathbf{s}, \mathbf{v})} \quad (2.161)$$

Způsob výpočtu Jacobiho matice $\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})$ a vektoru pravé strany rovnic kinematiky $\beta_c(\mathbf{s}, \mathbf{v})$ je zobrazen v rovnici (2.161).

Kapitola 3

Vytvoření softwarového nástroje pro automatické sestavení dynamického modelu tensegritické struktury

Hlavním cílem této práce bylo vytvořit univerzální software pro automatické sestavování pohybových rovnic pro 3D multibody systémy s pružinami, tlumiči, lany a sférickými vazbami. V této kapitole budou popsány specifikace vytvořeného softwaru a částečně způsob, jakým software pohybové rovnice sestavuje.

3.1 Vstupy softwaru

Pro zadání vyšetřované struktury je nejprve potřeba ji rozdělit na jednotlivá vyšetřovaná tělesa. Následně zvolit na tělesech body pevně s tělesem spojené (také označované jako *markery*). Posledním krokem je specifikace, jakým způsobem mají být body na tělesech spojené. Mezi možná spojení patří sférická vazba, spojení dvou bodů pružinou, tlumičem případně lanem. U lana je možné zadat více než dva přípojně body. Při zadání více než dvou bodů je uvažováno provlečené lano popsané v sekci 2.8.2.

Jednotlivé části struktury jsou zadávány do příslušných vstupních souborů formátu .CSV. Ten byl zvolen z důvodu možné editace v jednoduchých programech, jako je poznámkový blok, ale i možnost úpravy v Excelu.

3.1.1 Vstupy těles

Prvním popisovaným vstupním souborem je soubor *BODY.CSV*. Do něho se zapisují informace o simulovaných tělesech. Každý řádek souboru reprezentuje jedno těleso. Každému tělesu je třeba přiřadit následující vlastnosti:

- Unikátní ID tělesa
- Hmotnost
- Prvky matice setrvačnosti
- Počáteční podmínky
 - Poloha
 - Rychlost

Vzhledem k tomu, že pohybové rovnice jsou psané ke středu hmotnosti tělesa, je třeba vyplňovat vlastnosti těles vztažené právě k němu. Tato poznámka je důležitá hlavně při zadávání složek matice setrvačnosti a markerů, tedy bodů pevně spojených s tělesy.

3.1.2 Vstupy bodů

Soubor *POINT.CSV* v sobě obsahuje informaci o bodech (*markerech*), které jsou pevně připojené k jednotlivým tělesům nebo rámu. Ke každému vytvořenému bodu je potřeba vyplnit následující vlastnosti:

- Unikátní ID
- ID tělesa, ke kterému je bod pevně připojen (0 je vyhrazena pro rám)
- Poloha bodu vůči středu hmotnosti zapsaná v lokálním souřadnicovém systému

3.1.3 Vstupy lan, pružin a tlumičů

Pro přidání lan, pružin a tlumičů do softwaru se používají soubory *CABLE.CSV*, *SPRING.CSV* a *DAMPER.CSV*. Tyto tři vstupní soubory mají velice podobnou strukturu, a proto jsou zde popsány dohromady. Potřebné vlastnosti pro přidání těchto prvků jsou:

- Unikátní ID lana, pružiny nebo tlumiče
- ID bodů, ke kterým je daný prvek připojen
 - Lana mohou mít i více přípojných bodů
- Vlastnosti charakterizující prvek
 - Tuhost a volná délka u pružin a lan
 - Tlumení u tlumičů a lan

3.1.4 Vstupy kloubů

Pro spojení dvou bodů sférickou vazbou lze využít soubor *JOINT.CSV*. Potřebné parametry pro přidání této vazby do softwaru jsou:

- Unikátní ID sférické vazby
- ID bodů, které se mají sférickou vazbou spojit

3.2 Struktura softwaru

V softwaru se používá pojem stavový vektor označovaný jako \mathbf{x} . Ten se skládá z vektoru souřadnic \mathbf{s} a vektoru rychlostí \mathbf{v} . Obdobným způsobem je definována derivace stavového vektoru $\dot{\mathbf{x}}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{s} \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^{12n_B} \quad \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\mathbf{v}} \end{bmatrix} \in \mathbb{R}^{12n_B} \quad (3.1)$$

Software je psaný za pomoci objektově orientovaného programování (OOP). Jako hlavní třída slouží třída *TenseMech*, která slouží k interakci s uživatelem. Obsahuje následující nejdůležitější metody:

- *plotCurrentPosion(ax)* - vykreslení aktuální pozice systému
- *matlabSim(timeStop)* - simulace struktury v ODE řešiči v rámci prostředí MATLAB
- *animate(ax)* - vykreslení animaci odsimulované struktury

Jednoduchý kód pro základní práci se softwarem lze vidět na obr. 3.1. Ukázkový kód vytvoří novou instanci objektu *TenseMech*, vykreslí počáteční polohu struktury do grafu a vypočítá deset vteřin simulace. Poslední řádek kódu výsledek zobrazí ve formě animace uživateli.

3.2.1 Inicializace softwaru

Pro úspěšnou inicializaci respektive vytvoření nové instance třídy *TenseMech* je potřeba několik inicializačních výpočtů. Do nich patří načtení vstupních souborů, validace vstupních dat nebo výpočty konstant. Mezi nejdůležitější počítané konstanty patří matice hmotnosti všech těles $\mathbf{M}(\mathbf{s})$ a vektor tíhových účinků \mathbf{W} . Do konstant počítaných v této části patří i informace o počtu těles n_B a o počtu vazbových rovnic n_c , sloužící pro výpočet správných rozměrů matic. Akce vykonané po vytvoření instance jsou znázorněny v diagramu aktivit obr. 3.2.

```

%Vytvoření nové instance TenseMech
tm = TenseMech();

%Vytvoření grafu a následné vykreslení počáteční
  konfigurace
f = figure; axf = gca; cla(axf); hold(axf, 'on');
tm.plotCurrentPosition(axf);

%Odsimulování 10 vteřin
tm = tm.matlabSim(10);

%Vykreslení simulace do grafu v podobě animace
tm.animate(axf);

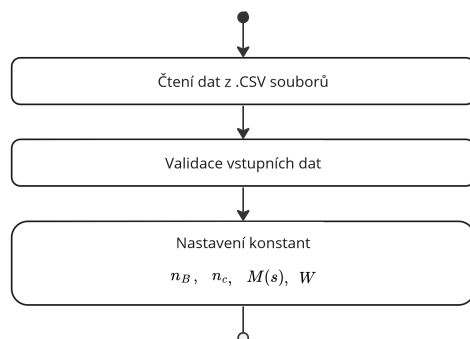
```

Obr. 3.1: Ukázka práce se softwarem

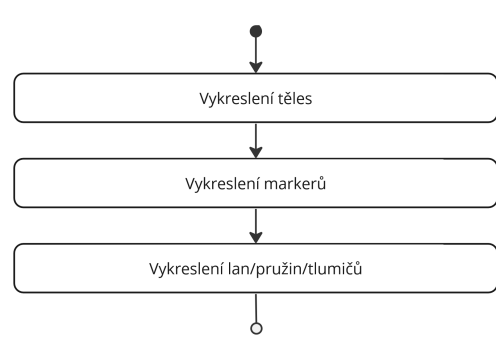
3.2.2 Vykreslení aktuální polohy

Důležitou funkcí vytvořeného softwaru je možnost vykreslit aktuální polohu struktury. Ta je velmi užitečná při kontrole zadaných vstupů uživatelem. Díky grafickému znázornění vstupních dat může uživatel upravovat počáteční konfiguraci struktury. Kromě toho je důležité mít možnost kontrolovat i které uzly jsou mezi sebou propojeny lany, tlumiči a pružinami. To vše zajišťuje metoda *plotCurrentPosition(ax)* obsažená v třídě *TenseMech*. Metoda má jako povinný vstupní argument objekt Axes. Jednoduše řečeno je nutné specifikovat, do kterého grafu se má počáteční poloha vykreslit.

Postup vykreslování je schématicky zachycen v diagramu aktivit pro metodu *plotCurrentPosition(ax)* obr. 3.3. Po zavolání metody se vykreslí nejprve všechna tělesa a následně všechny markery. Posledním krokem je do vykreslovaného grafu přidat všechna lana, pružiny a tlumiče.

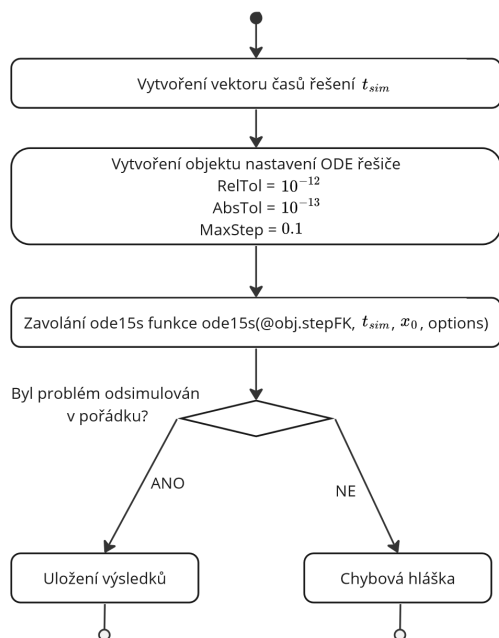


Obr. 3.2: Diagram aktivit inicializace objektu *TenseMech*

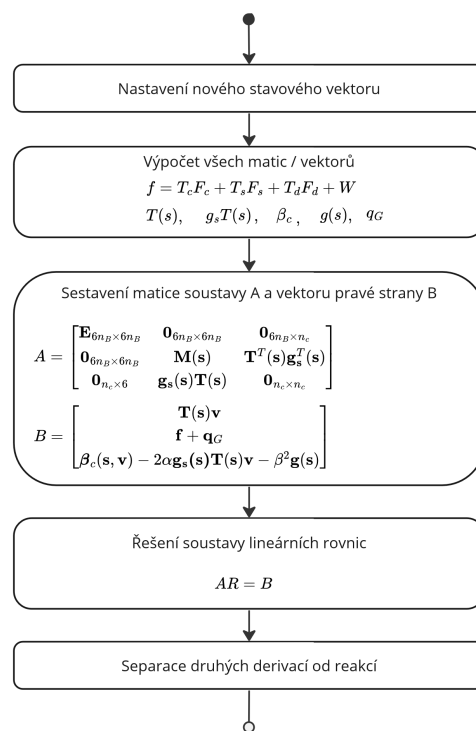


Obr. 3.3: Diagram aktivit vykreslování aktuální polohy struktury

3.2.3 Simulace dynamiky struktury



Obr. 3.4: Diagram aktivit metody `matlabSim(timeStop)`



Obr. 3.5: Diagram aktivit metody `stepFK(t, x)`

Integrace pohybových rovnic popisujících dynamiku zadané struktury je provedena za použití připravených funkcí v prostředí MATLAB. Jako výchozí nastavený řešič (ang. *solver*) byl zvolen ODE15s, vhodný pro řešení numerických úloh označovaných jako *stiff*. Tento druh řešiče se ukázal být nejvhodnější při simulaci struktur ve vytvářeném softwaru. Hlavní výhodou řešiče přináší v úspoře výpočetního času. V případech, kdy jsou všechna lana plně natažena s dostatečně velkou deformací, je možné provádět relativně velké integrační kroky.

V případech, kdy je deformace lana téměř nulová, může docházet k neustálému propínání a prověšování lana. Běžně používané Rungeovy–Kuttovy metody nejsou vhodné pro řešení stiff problémů a jejich použití může vést k velmi malým integračním krokům respektive velkým časům řešení úlohy.

Stiff solver ODE15s dokáže efektivně měnit velikost svého integračního kroku, a tím pádem v oblastech, kdy se řešení příliš nemění, zvýšit krok, a tím snížit výpočetní čas. Naopak v oblastech stavového prostoru, kde dochází k rychlým změnám, dokáže časový krok snížit a zachytit v řešení simulovanou dynamiku.

Schématicky naznačený postup řešení dynamiky je k vidění na obr. 3.4 a obr. 3.5. První obr. znázorňuje akce softwaru při zavolání metody `matlabSim(timeStop)`, použité v ukázkovém kódu obr. 3.1. V diagramu je vidět, že po zavolání metody se vytvoří vektor časů t_{sim} ve kterém budou výsledky simulace uloženy. Není žádoucí, aby byly uchovány všechny integrační časy použité v solveru, protože uchovávat

řešení o velikosti např. 10^5 prvků nepřináší téměř žádné výhody a naopak zabírá velice mnoho místa v paměti RAM.

Druhým krokem je vytvoření objektu nastavení solveru. Na obr. 3.4 je v druhém okně vidět, že hlavní parametry nastavení řešiče jsou *RelTol*, *AbsTol* a *MaxStep*. Zde uživatel může řídit přesnost a rychlost řešení, případně může zachovat základní nastavení.

Následuje zavolání řešiče. Přednastavenou volbou je ODE15s, tedy solver vhodný pro úlohy typu stiff. MATLAB ovšem umožňuje rychlou změnu použitého řešiče pomocí jednoduchého nahrazení funkce ODE15s za libovolný jiný řešič. Např. pokud uživatel potřebuje velice rychlé řešení za cenu menší přesnosti, je možné ODE15s nahradit ODE23. Při praktickém zkoušení jednotlivých solverů se ODE15s ukázal být nejlepší volbou a proto je i zde nastaven jako základní volba.

Funkce, která počítá derivaci stavového vektoru, se jmenuje *stepFK(t,x)*. Její diagram aktivit je k vidění na obr. 3.5. Tato metoda je použita jako první argument funkce ODE15s. Druhým argumentem je vektor integračních časů t_{sim} vytvořený v prvním kroku. Třetí argument je počáteční stavový vektor x_0 , obvykle složený z počátečních podmínek definovaných v souboru *BODY.CSV* viz subsekcce 3.1.1. Posledním argumentem je objekt nastavení solveru, zde již diskutovaný v druhém kroku.

Pokud ODE15s úspěšně dokončí simulaci, výsledky se uloží do vlastností třídy pod jmény t_{sim} a Y_{sim} . Může se ovšem stát, že simulace není úspěšná z důvodu singulární matice systému nebo z důvodu nedodržení vazbových rovnic. V takovém případě je uživateli vrácena chybová hláška a je nutné podrobněji analyzovat příčinu.

Již zmíněný druhý diagram zobrazený na obr. 3.5 znázorňuje akce při zavolání metody *stepFK(t, x)*, která počítá derivaci stavového vektoru. Jeho výpočet vychází z odvozené rovnice dynamiky se stabilizací (2.128). V prvním kroku je potřeba sestavit všechny dílčí matice a vektory pro aktuální stav struktury. Do nich patří:

- $\mathbf{T}(\mathbf{s})$ Matice přepočtu vektoru rychlostí \mathbf{v} na derivaci vektoru souřadnic $\dot{\mathbf{s}}$
- $\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})$ Jacobiho matice vazbových rovnic
- $\beta_c(\mathbf{s}, \mathbf{v})$ Matice pravé strany kinematických rovnic
- \mathbf{f} Vektor vnějších sil, obsahující síly v lanech, pružinách, tlumičích a účinky tíhy \mathbf{W}
- $\mathbf{g}(\mathbf{s})$ Chyba vazbových rovnic (pro stabilizaci)
- \mathbf{q}_G Gyroskopické účinky

Podrobnější popis výpočtu dynamiky soustavy těles byl popsán v sekci 2.7 s definicí matic pro jednotlivá tělesa v sekci 2.5.

Matice hmotnosti $\mathbf{M}(\mathbf{s})$ a vektor tíhových sil \mathbf{W} jsou konstantní a počítají se při inicializaci instance třídy `TenseMech` popsané v subsekcí 3.2.1.

Po sestavení dílčích matic a vektorů se sestaví celková matice systému \mathbf{A} a vektor pravé strany \mathbf{B} . Pomocí nich se řešením soustavy lineárních rovnic $\mathbf{AR}=\mathbf{B}$ získá řešení \mathbf{R} , které obsahuje derivaci stavového vektoru a Lagrangeovy multiplikátory.

$$\mathbf{R} = \begin{bmatrix} \dot{\mathbf{x}} \\ -\boldsymbol{\lambda} \end{bmatrix} \quad (3.2)$$

Metoda vrátí odděleně vektor derivace a multiplikátorů. Samotné řešení dynamiky soustavy těles nepotřebuje multiplikátory (reakční účinky). Je ovšem možné metodu použít zpětně tak, že se do ní dosazují stavové vektory a získávají se pouze reakce. Tak je možné po simulaci efektivně získat průběh multiplikátorů bez nutnosti uchovávat je při samotné simulaci.

3.2.4 Výsledky simulace

Po dokončení simulace struktury je možné vykreslit průběhy souřadnic a rychlostí v jednotlivých časových okamžicích. K tomu je vytvořena metoda `plotCoordinates(ax, coordinates)`.

```
% Metoda pro vykreslení souřadnic x, y, z prvního tělesa  
f = figure; axf = gca; cla(axf); hold(axf, 'on');  
tm.plotCoordinates(axf, [1, 2, 3]);
```

Obr. 3.6: Příkaz pro vykreslení souřadnic x, y, z prvního tělesa

Jako příklad je na obr. 3.6 zobrazen příkaz pro vykreslení průběhu souřadnic x, y, z prvního tělesa. Obdobným způsobem lze vykreslit libovolnou souřadnici ze stavového vektoru.

3.2.5 Animace výsledku

Kromě vykreslení výsledků simulace v podobě grafů lze výsledek zobrazit i ve formě animace. Pro spuštění animace je připravena metoda `animate(ax)`. Použití této metody je zobrazena na obr. 3.1.

Rozšířený návod pro ovládání softwaru je dostupný v příloze A. Samotný software je k dispozici v přiloženém souboru typu ZIP (příloha B).

Kapitola 4

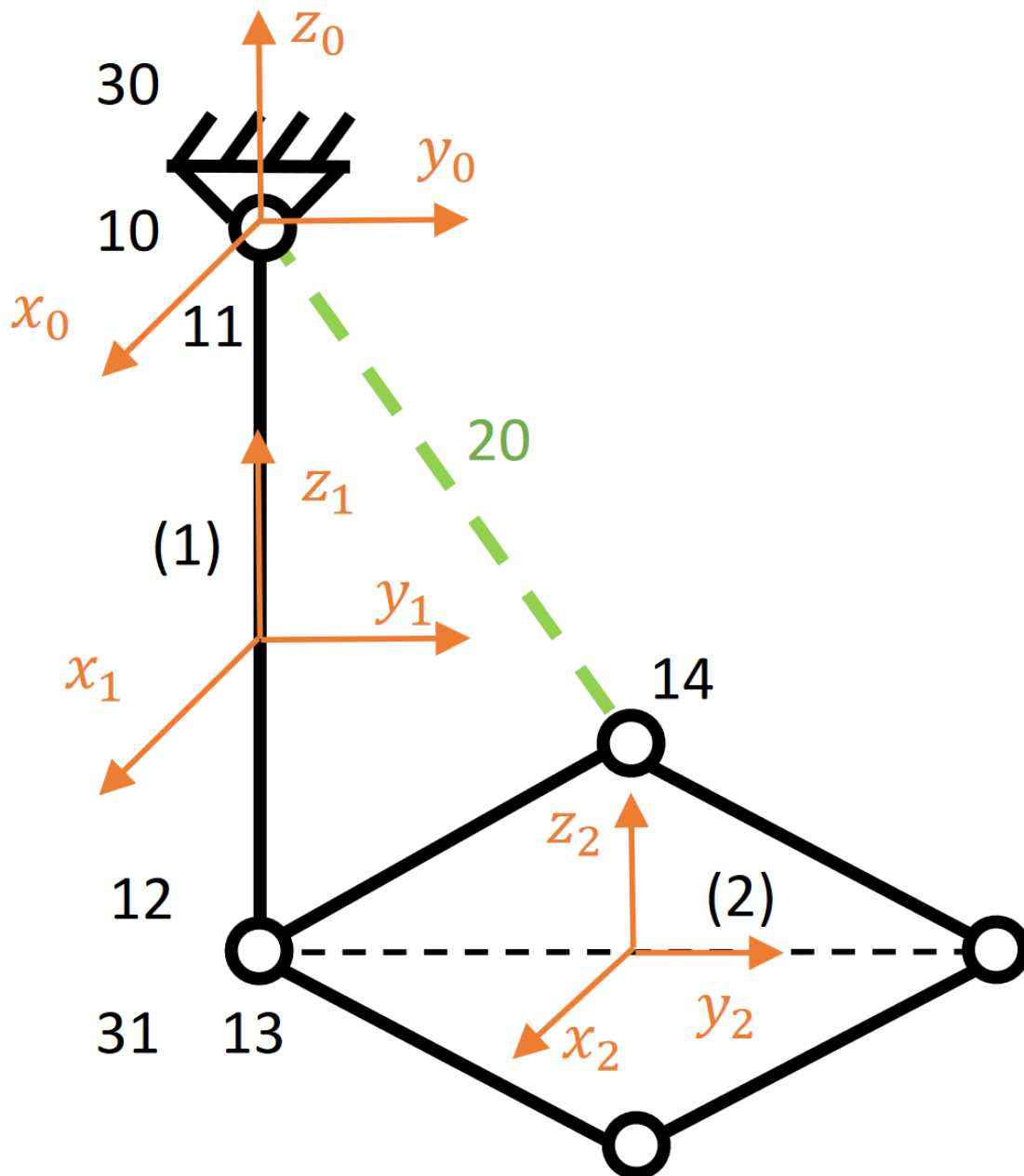
Ověření vytvořeného modelu vybrané tensegritické struktury s modelem vytvořeným v prostředí MATLAB-SIMULINK- SIMSCAPE

V této kapitole budou prezentovány výsledky dynamických simulací modelů vzniklých pomocí vytvořeného softwarového nástroje. Výsledky budou porovnány s modelem vytvořeným v prostředí SIMSCAPE MBS. Pro porovnání výsledků byly zvoleny dva příklady. Prvním z nich je model modifikovaného dvojkyvadla řešeného v sekci 4.1. Druhým porovnávaným modelem je struktura složená ze tří tyčí a desky, blíže popsaná v sekci 4.2.

4.1 Dvojkyvadlo

První strukturou pro verifikaci softwaru je model modifikovaného dvojkyvadla. V této sekci budou popsány vstupní soubory vytvořeného softwaru pro model dvojkyvadla. Bude porovnána složitost vstupů vytvořeného softwaru vůči prostředí SIMSCAPE. Na závěr budou porovnány výsledky dynamické simulace.

Řešený problém je vyobrazen na obr. 4.1. Dvojkyvadlo se skládá ze dvou tuhých těles zobrazených černou barvou s identifikátory 1 a 2. Prvním z těles je 2 m dlouhá tyč. Ta je připojena k rámu pomocí sférické vazby. Druhým tělesem je čtvercová deska s délkou úhlopříčky 2 m. Obě tělesa jsou spolu spojena sférickou vazbou. Jeden z rohů desky je k rámu připojen lanem.



Obr. 4.1: Náčrt řešeného modelu dvojkyvadla s lanem

4.1.1 Vstupní soubory softwaru

Pro vytvoření modelu dvojkyvadla je zapotřebí vyplnit následující vstupní soubory *BODY.CSV*, *POINTS.CSV*, *CABLE.CSV* a *JOINTS.CSV*. Data, která byla třeba do souborů vyplnit, jsou zobrazena v tabulkách 4.1, 4.2, 4.3 a 4.4. Podrobnější informace o vstupech byly diskutovány v kapitole 3.

V tabulce 4.1 jsou zobrazena vstupní data souboru *BODY.CSV*. Dva řádky tabulky reprezentují dvě tělesa (ID 1 a 2). První těleso je tyč o hmotnosti 30 *kg*. Její tenzor setrvačnosti je složen pouze z diagonálních prvků o velikosti 10, 10 a 0.01 *kgm²* (jedná se o hlavní osy tenzoru setrvačnosti). Počáteční poloha je určena z obr. 4.1. Souřadnicový systém tyče je vůči souřadnicovému systému rámu posunut pouze ve směru osy *z* o -1 *m* a není vůči němu natočený. Počáteční rychlosti tyče jsou nulové.

Druhým tělesem je čtvercová deska o hmotnosti 350 *kg*. Její souřadnicový systém je v počáteční poloze posunut vůči počátku rámu o 1 *m* ve směru osy *y* a o -2 *m* ve směru osy *z*. Její počáteční rychlost je nulová.

ID	Mass	I_x	I_y	I_z	D_{xy}	D_{xz}	D_{yz}	x_0	y_0	z_0	ϕ_{x0}	ϕ_{y0}	ϕ_{z0}	v_{x0}	v_{y0}	v_{z0}	ω_{x0}	ω_{y0}	ω_{z0}
1	30	10	10	0.01	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
2	350	30	30	60	0	0	0	0	1	-2	0	0	0	0	0	0	0	0	0

Tabulka 4.1: Dvojkyvadlo - *BODY.CSV*

Druhou tabulkou je tabulka uzlů (tab. 4.2). V ní jsou uloženy lokální rádiusvektory těles (*markery*). První sloupec obsahuje ID uzlů. Ve druhém sloupci je k nalezení ID tělesa, ke kterému je uzel pevně připojen. Bod 10 je pevně spojen s rámem. Souřadnicový systém rámu je označen číslem 0. Body 11 a 12 jsou pevně spojeny s tyčí. Body 13 a 14 jsou pevně přichyceny k desce. Markery s ID 15 a 16 jsou do tabulky dodány z důvodu vizualizace. Kdyby nebyly do tabulky přidány, deska by se nevykreslovala správně. Všechny body jsou vyjádřeny v souřadnicových systémech tělesa, ke kterému jsou přichycena. Tabulka 4.2 zobrazuje vstupní data souboru *POINT.CSV*.

ID	IDCON	x	y	z
10	0	0	0	0
11	1	0	0	1
12	1	0	0	-1
13	2	0	-1	0
14	2	-1	0	0
15	2	0	1	0
16	2	1	0	0

Tabulka 4.2: Dvojkyvadlo - *POINT.CSV*

Pro zadání sférických vazeb slouží tabulka *JOINT.CSV*. Její obsah je k vidění

v tabulce 4.3. Pro model dvojkyvadla bylo potřeba dvou sférických vazeb. První z nich modeluje spojení tyče s rámem. Kloubová vazba spojuje uzly 10 a 11. Druhou sférickou vazbu lze nalézt mezi tyčí a deskou a to mezi uzly 12 a 13.

ID	ID_FROM	ID_TO
50	10	11
51	12	13

Tabulka 4.3: Dvojkyvadlo - JOINT.CSV

Posledním vstupním souborem je soubor CABLE.CSV. Ten obsahuje definici použitých lan. V případě modelovaného dvojkyvadla obsahuje zadání jednoho lana s ID 20 spojujícího uzly 14 a 11¹. Měrná tuhost lana je $10000 \frac{mN}{m}$ a tlumení $100 \frac{mNs}{m}$. Volná délka lana je 1.5 *m*. Informace o laně jsou vypsány v tabulce 4.4.

ID	k	b	l_0	ID_THROUGH
20	10000	100	1.5	11,14

Tabulka 4.4: Dvojkyvadlo - CABLE.CSV

Spuštění simulace

Dalším krokem po vyplnění vstupních souborů je spuštění simulace. Tento proces byl již popsán v kapitole 3 přesněji na obr. 3.1.

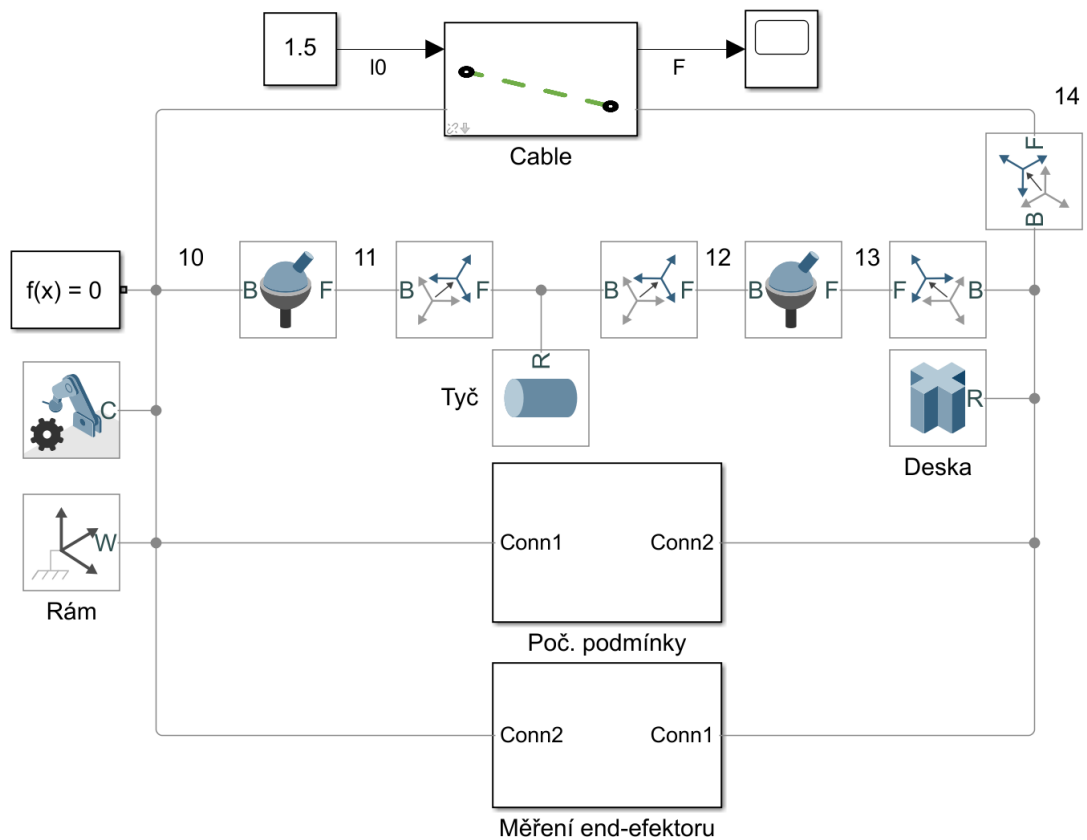
4.1.2 Model dvojkyvadla v prostředí SIMSCAPE

Za účelem verifikace vytvořeného softwaru bylo nutné sestavit model v prostředí SIMSCAPE. Na obr. 4.2 je zobrazen výsledný model. Simulační model se skládá ze dvou tuhých těles označených jako *Tyč* a *Deska*. Model obsahuje blok *Cable*, který byl popsán a použit v práci [8]. Subsystém *Poč. podmínky* nastavuje desce její počáteční polohu v prostoru. Subsystém *Měření* ukládá výsledky simulace do *Workspace*. Na obr. 4.2 jsou dále očíslovány jednotlivé spoje reprezentující ID markery z obr. 4.1.

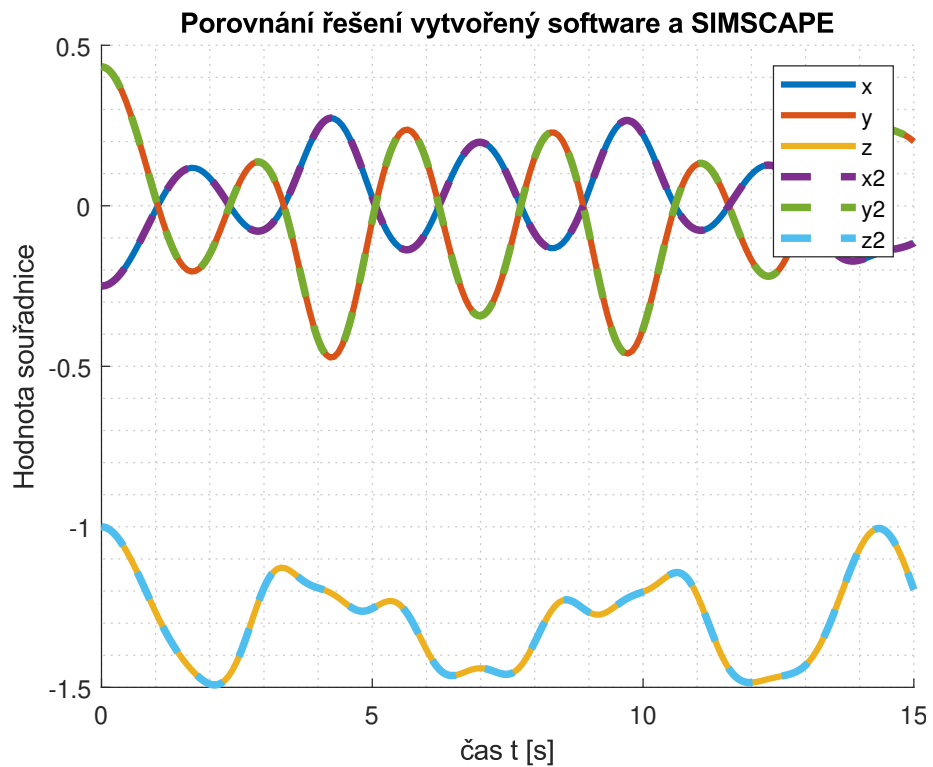
4.1.3 Porovnání modelů dvojkyvadla

Pro porovnání výsledků z vytvářeného softwaru s výsledkem z prostředí SIMSCAPE MBS bude použito průběhu translačních souřadnic x, y, z středu hmotnosti desky. Na obr. 4.3 lze vidět jejich průběh v čase. Plnou čarou jsou vykresleny výsledky simulace z vytvářeného softwaru. Čárkovanou čarou jsou znázorněny výsledky z prostředí SIMSCAPE MBS. Výsledky při tomto pohledu vypadají nerozlišitelně.

¹případně uzly 14 a 10, není rozdíl mezi těmito dvěma vstupy

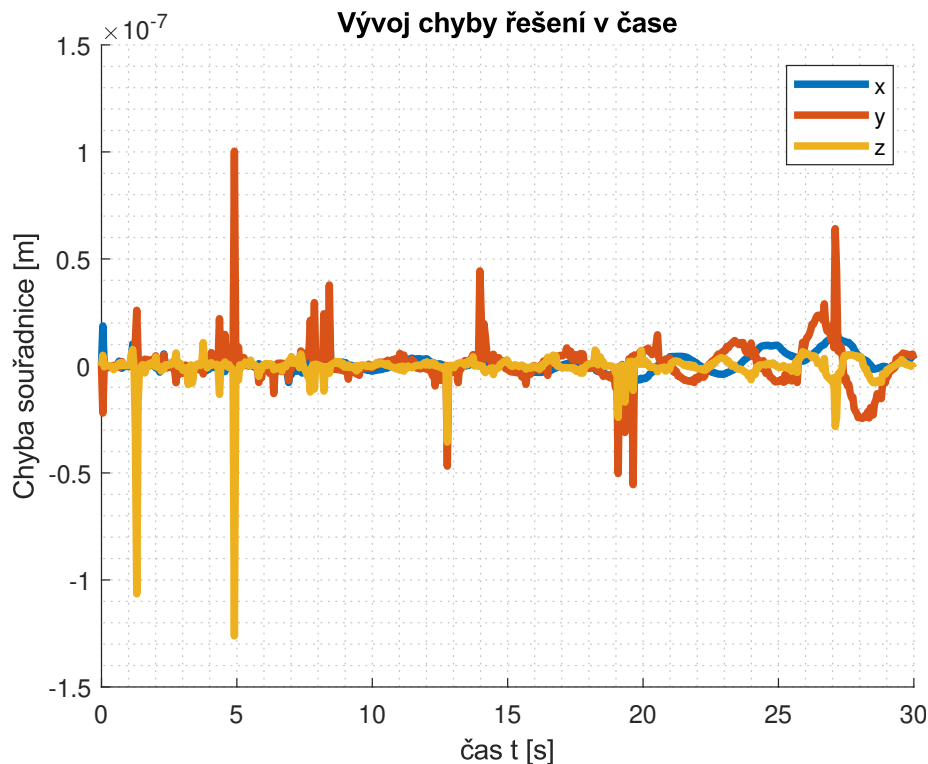


Obr. 4.2: Model v prostředí SIMSCAPE MBS pro dvojkyvadlo



Obr. 4.3: Porovnání výsledků simulace dvojkyvadla

Pro hlubší porovnání výsledků byly průběhy souřadnic z vytvořeného softwaru odečteny od výsledků z prostředí SIMSCAPE MBS. Výsledný rozdíl (absolutní chyba souřadnice) je zobrazen na obr. 4.4.



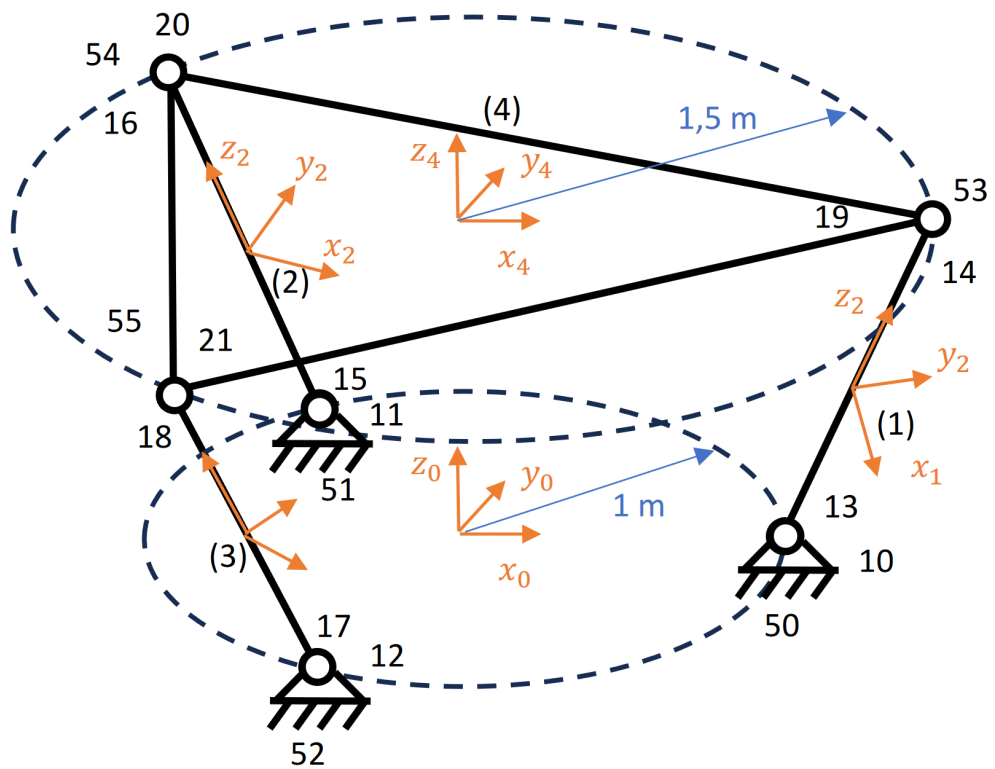
Obr. 4.4: Průběh absolutní chyby simulace dvojkyvadla

Je vidět, že se chyba ve 30 s dlouhé simulaci drží řádově $10^{-7}m$. Tento výsledek lze považovat za dostatečný pro verifikaci modelu.

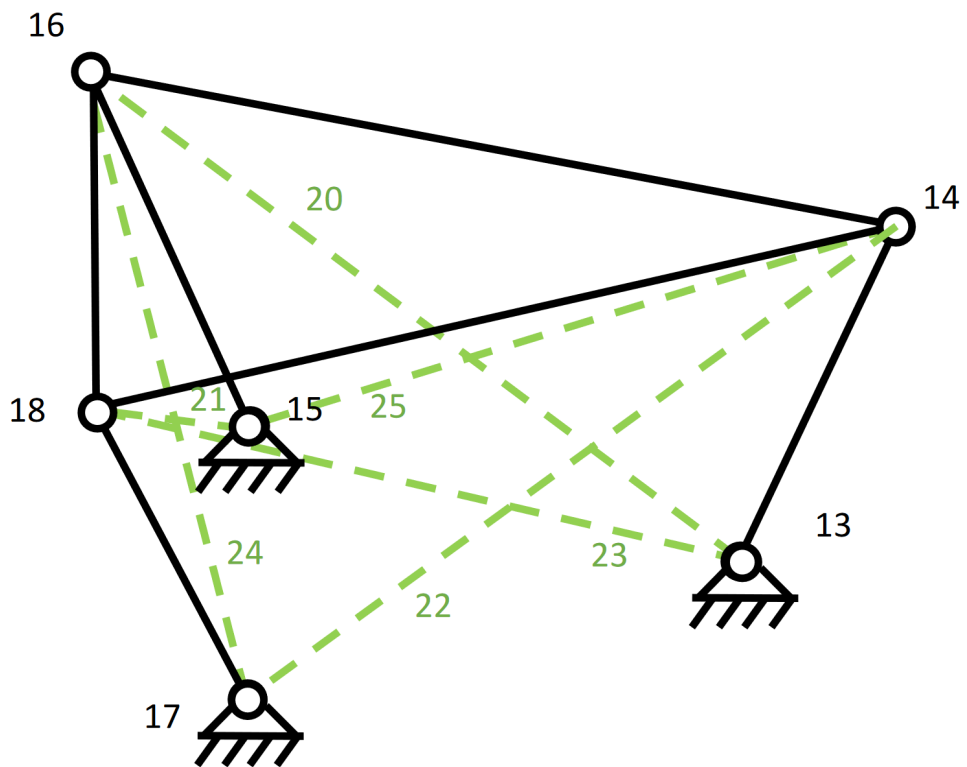
4.2 Platforma

Druhou testovanou strukturou je struktura tvořená ze tří tyčí a desky. Dále bude tato struktura označovaná jako *platforma*. Schéma struktury je zobrazeno na obr. 4.5 a obr. 4.6. Na obr. 4.5 jsou zobrazeny 3 tyče, jednou stranou sférickou vazbou spojené k rámu, označené čísla 1, 2 a 3. Číslem 4 je označena deska s tyčemi spojená také pomocí sférických vazeb. Každé z těles má svůj lokální souřadnicový systém označený číslem tělesa. Tyče jsou rovnoměrně rozmístěny na kružnici o poloměru 1 m a jsou dlouhé 2 m. Deska má tvar rovnostranného trojúhelníku vepsaného do kružnice o velikosti 1.5 m. Na tomto prvním schématu nebyla zakreslena lana z důvodu přehlednosti.

Na obr. 4.6 je vidět ta samá struktura zakreslená s lany. Ve struktuře bylo použito 6 shodných lan označených čísly 20 až 25.



Obr. 4.5: Schéma 1 platformy



Obr. 4.6: Schéma 2 platformy

4.2.1 Vstupní soubory softwaru

Pro zadání struktury do vytvořeného softwaru bylo zapotřebí použití stejných vstupních souborů jako tomu bylo u modelu platformy. V této sekci budou uvedeny pouze tabulky reprezentující vstupní soubory bez bližšího popisu. Bližší popis vstupních souborů je k dispozici v kapitole 3 nebo v subsekcí 4.1.1. Uvedené tabulky obsahují zaokrouhlené vstupní údaje z důvodu přehlednosti tabulek. Pro reálné simulace bylo použito nezaokrouhlených dat. Tabulky s daty jsou zde uvedeny pod čísly 4.5, 4.6, 4.7 a 4.8.

ID	Mass	I_x	I_y	I_z	D_{xy}	D_{xz}	D_{yz}	x_0	y_0	z_0	ϕ_{x0}	ϕ_{y0}	ϕ_{z0}	v_{x0}	v_{y0}	v_{z0}	ω_{x0}	ω_{y0}	ω_{z0}
1	1.00	1.00	1.00	0.10	0	0	0	1.25	0.00	0.97	0.00	0.25	0.00	0	0	0	0	0	0
2	1.00	1.00	1.00	0.10	0	0	0	-0.62	1.08	0.97	-0.22	-0.13	2.08	0	0	0	0	0	0
3	1.00	1.00	1.00	0.10	0	0	0	-0.62	-1.08	0.97	0.22	-0.13	-2.08	0	0	0	0	0	0
4	5.00	1.00	1.00	5.00	0	0	0	0	0	1.94	0	0	0	0	0	0	0	0	0

Tabulka 4.5: Platforma - BODY.CSV

ID	IDCON	x	y	z
10	0	1.00	0.00	0.00
11	0	-0.50	0.87	0.00
12	0	-0.50	-0.87	0.00
13	1	0.00	0.00	-1.00
14	1	0.00	0.00	1.00
15	2	0.00	0.00	-1.00
16	2	0.00	0.00	1.00
17	3	0.00	0.00	-1.00
18	3	0.00	0.00	1.00
19	4	1.50	0.00	0.00
20	4	-0.75	1.30	0.00
21	4	-0.75	-1.30	0.00

Tabulka 4.6: Platforma - POINT.CSV

ID	ID_FROM	ID_TO
50	10	13
51	11	15
52	12	17
53	14	19
54	16	20
55	18	21

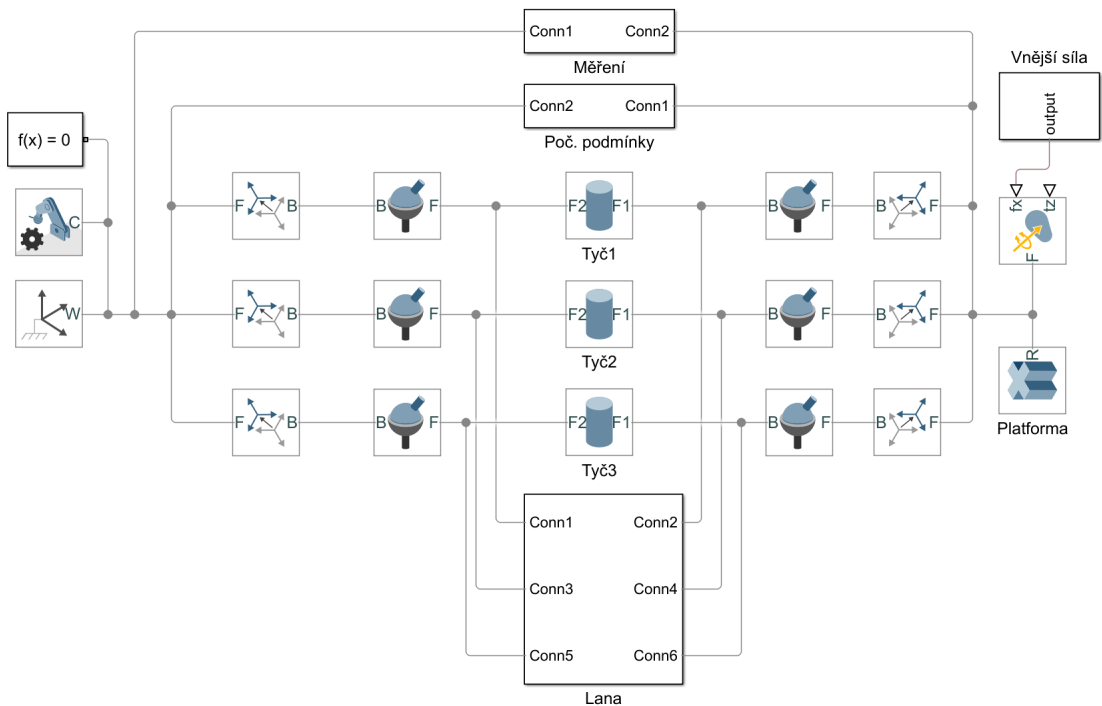
Tabulka 4.7: Platforma - JOINT.CSV

ID	k	b	l_0	ID_THROUGH
20	1500	10	2.91	13,16
21	1500	10	2.91	15,18
22	1500	10	2.91	17,14
23	1500	10	2.91	13,18
24	1500	10	2.91	17,16
25	1500	10	2.91	15,14

Tabulka 4.8: Platforma - CABLE.CSV

4.2.2 Model platformy v prostředí SIMSCAPE

Stejně jako v případě dvojkyvadla i u modelu platformy bylo potřeba vytvořit model v prostředí SIMSCAPE MBS za účelem verifikace vytvářeného softwaru. Výsledný model v prostředí SIMSCAPE MBS je zobrazen na obr. 4.7. Na něm lze vidět model složený ze tří tyčí připojených sférickou vazbou k rámu. Subsystém *Lana* obsahuje všechny modely lan, které pro přehlednost modelu musely být umístěny do samostatného subsystému. Druhá část modelu obsahuje sférické vazby spojující tyče s deskou (označené v modelu jako *Platforma*). V modelu je umístěna i vnější síla, jejíž funkce bude popsána v následující subsekcí 4.2.3. Poslední dva subsystémy jsou subsystém *Měření* a *Poč. podmínky*. *Měření* obsahuje bloky snímající polohu desky, která je následně ukládána do *Workspace*. Subsystém *Poč. podmínky* nastavuje počáteční polohu desky.

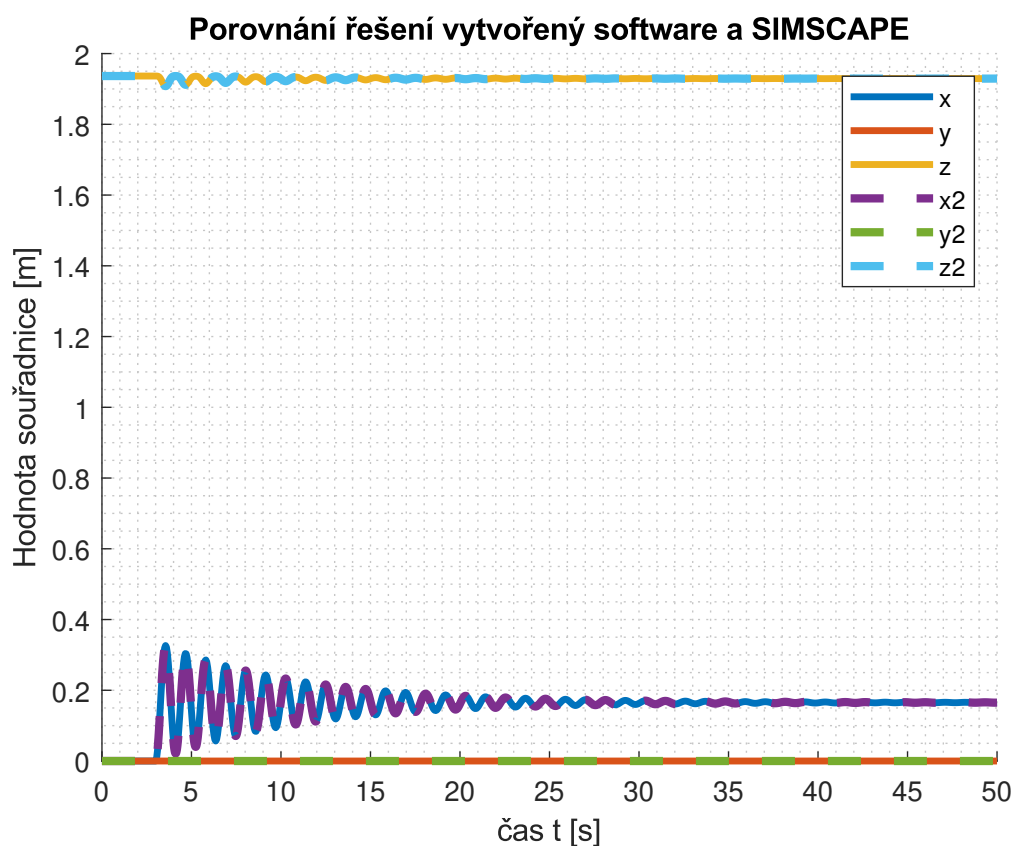


Obr. 4.7: Model v prostředí SIMSCAPE MBS pro platformu

4.2.3 Porovnání modelů platformy

V této subsekcí budou porovnány výsledky simulací z vytvořeného softwaru a z prostředí SIMSCAPE MBS pro model platformy. Oběma modelům jsou nastaveny stejné počáteční podmínky, hmotností charakteristiky těles, stejné tíhové zrychlení a navíc je přidána vnější síla působící ve středu hmotnosti desky. Tato síla je modelována pomocí skokové změny ve třetí vteřině simulace. Síla působí ve směru kladné osy x . Byly provedeny dvě simulace. V první simulaci má síla velikost 40 N . Druhá simulace je při pouhých 5 N . Důvodem volby dvou sil je ukázka funkčnosti nelinearity lan. Lano je nelineární z důvodu nemožnosti přenosu tlakové síly. Pokud by lano mělo být tlačeno, nepřenese žádnou sílu. Stejně tomu tak je při velkých rychlostech, které by měly způsobit, že lano tlakovou sílu přenést umí. Podmínky pro vypínání síly v lanech byly popsány v subsekcí 2.8.5.

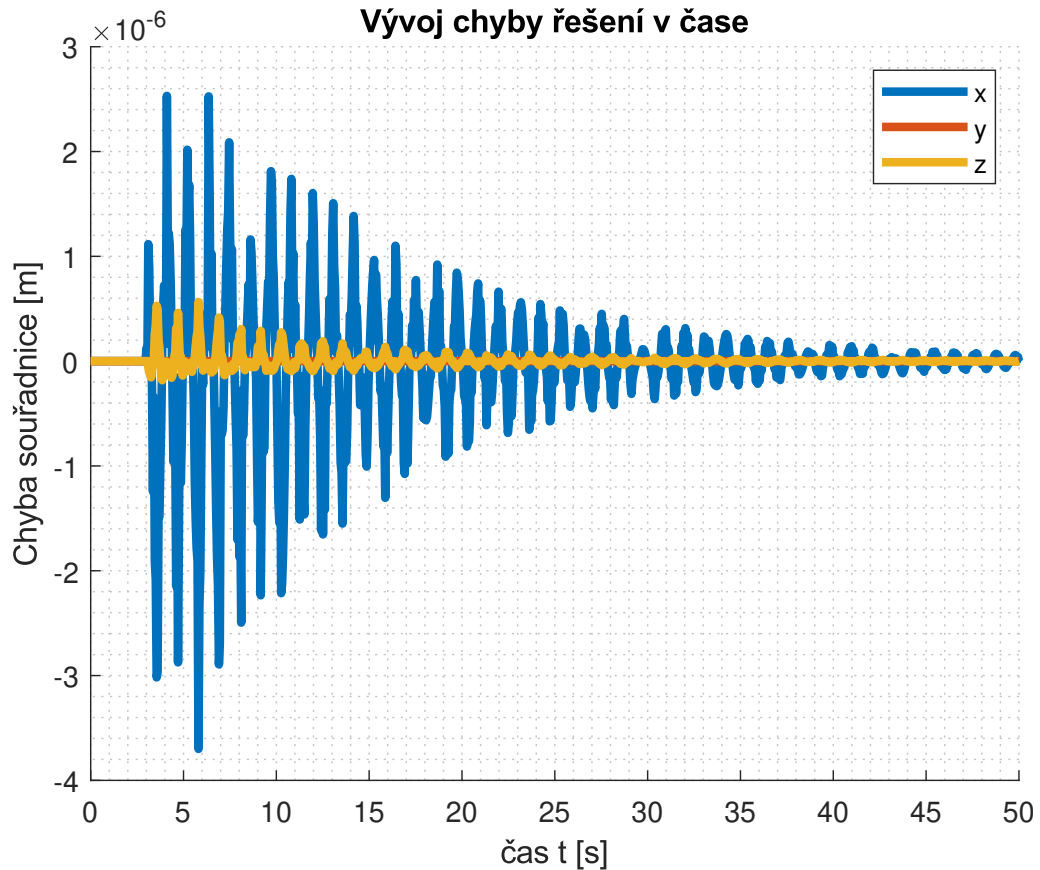
Na obr. 4.8 lze vidět průběhy souřadnic hmotného středu desky v čase. Ve třetí vteřině dojde ke skokovému zatížení desky silou o velikosti 40 N . Díky modelovanému tlumení lan model začne postupně zpomalovat a dosahovat nové rovnovážné polohy. Průběhy souřadnic vypočítané ve vytvořeném softwaru jsou zakresleny plnými čarami. Průběhy vypočítané v prostředí SIMSCAPE MBS jsou vykreslené čárkovaně. Pouhým okem jsou průběhy souřadnic neodlišitelné.



Obr. 4.8: Simulace platformy při síle 40 N

Pro hlubší analýzu byl vypočítán rozdíl mezi souřadnicemi ve všech časech. Průběh chyby lze vidět na obr. 4.9. Po skokové změně velikosti síly ve 3. vteřině chyba

vzroste na hodnotu $3 \cdot 10^{-6}$. Chyba s přibývajícím časem klesá. Tato chyba může být způsobena odlišným řešením detekce ztráty síly v laně. Modely lan využívají bloku saturace, který může používat jiné algoritmy detekce tzv. *zero crossing* než pouhý integrátor ODE15s. Vzniklá chyba je velmi malá a o softwaru lze říct, že i tuto úlohu správně vypočítal.

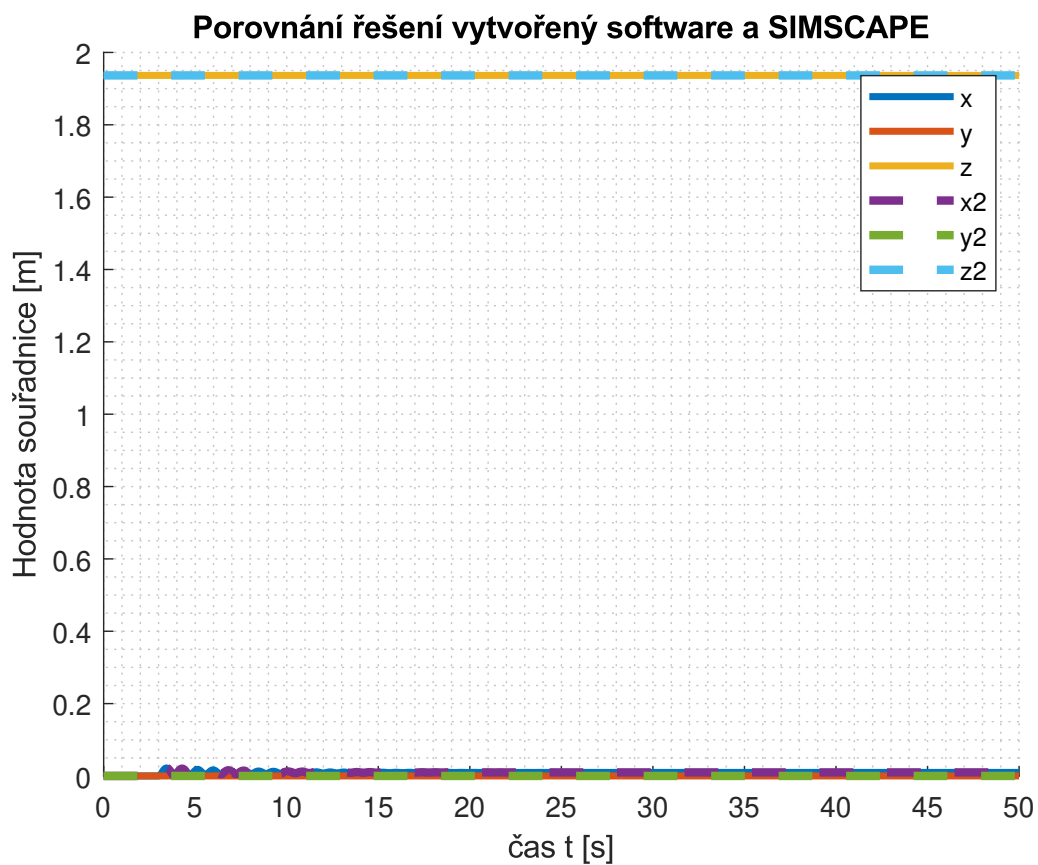


Obr. 4.9: Chyba průběhu souřadnic simulace platformy při síle 40 N

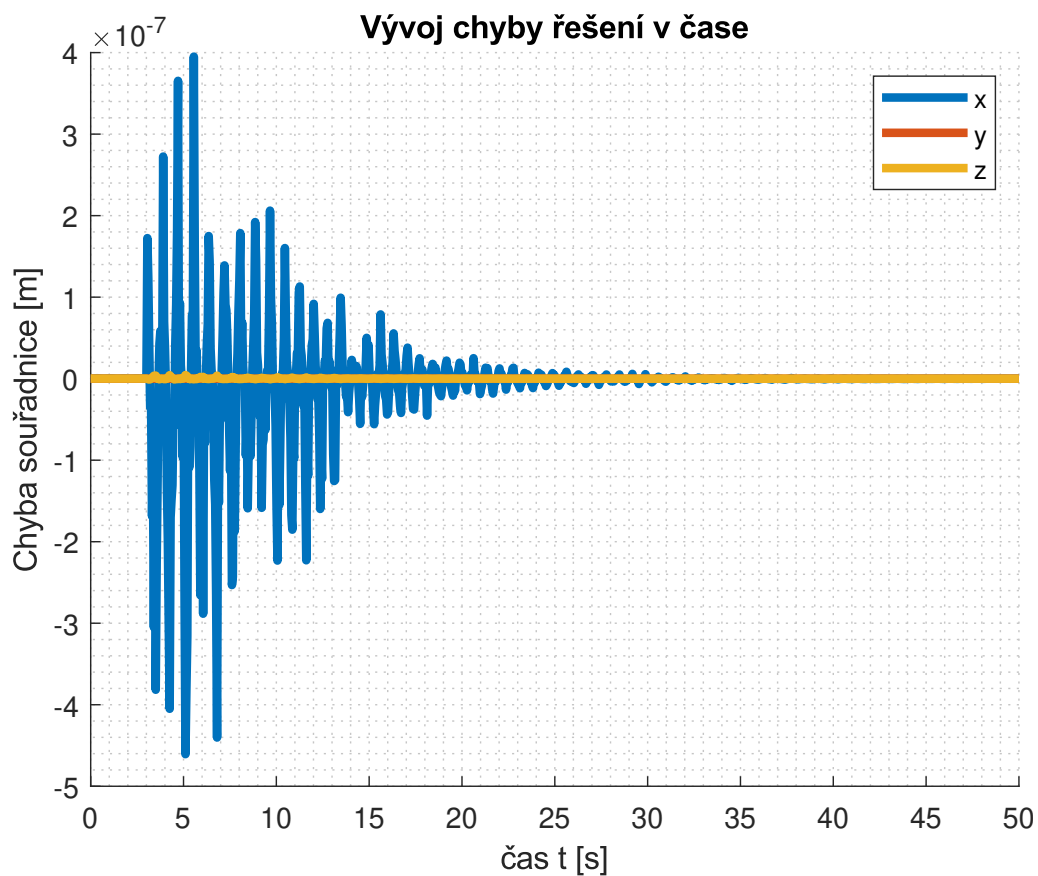
Druhá testovací simulace využívá sílu o velikosti 5 N. Průběh souřadnic desky je k vidění na obr. 4.10. Síla o takto malé velikosti se strukturou téměř nehnula. Výchytky jsou tedy velmi malé a není toho příliš vidět. Důležitý je graf velikosti chyb.

Absolutní velikost rozdílu mezi souřadnicí vypočítanou vytvářeným softwarem a prostředím SIMSCAPE MBS je zobrazen na obr. 4.11. Na něm je vidět, že modely jsou shodné i při zatížení silou o velikosti 5 N. Velikost chyby $4 \cdot 10^{-7}$ je dokonce menší než v případě zatížení silou o velikosti 40 N.

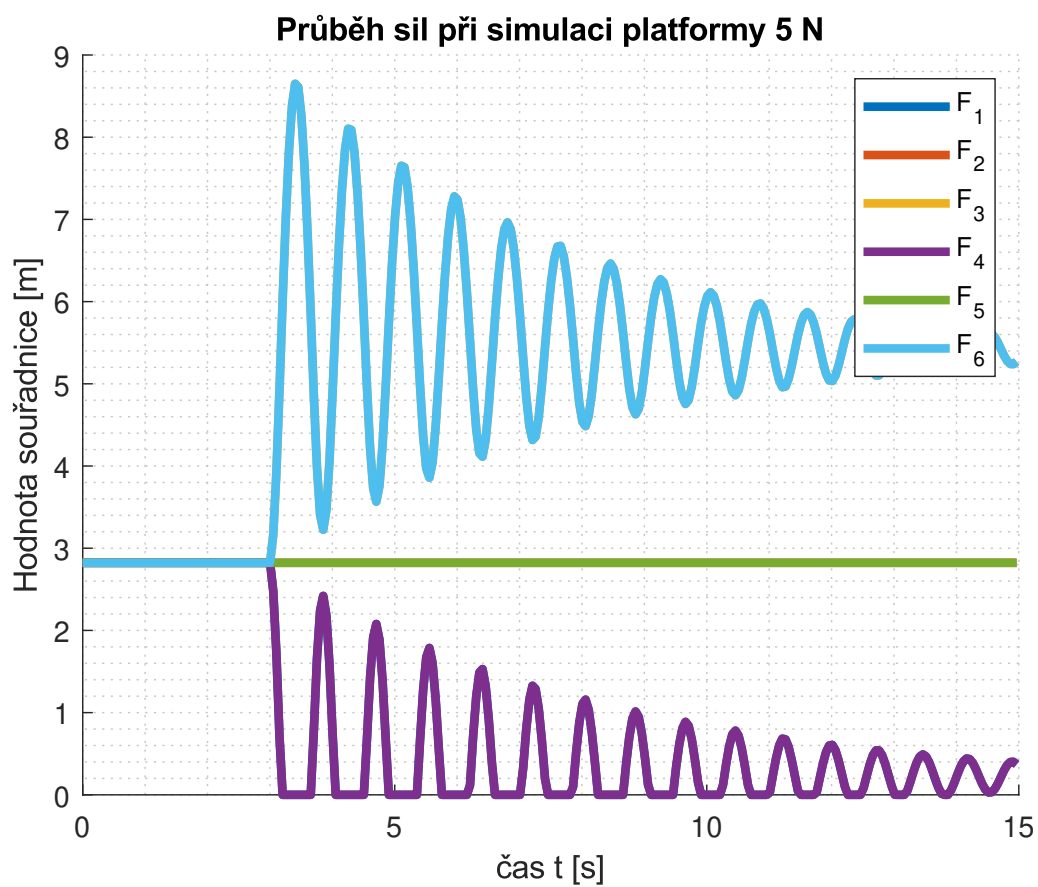
Hlavní důvod, proč je druhá simulace se silou 5 N uvedena, je, že se model po třetí vteřině simulace dostává velmi rychle a často do stavu ztráty síly v laně. Saturace velikosti sil je zobrazena na obr. 4.12. Na něm je vidět, že síly v lanech se rychle za sebou dostávají na hodnotu 0. Výsledky z obr. 4.11 pak znamenají, že vytvořený software využívající řešič ODE15s je efektivně schopný zvládnout situace, kdy dochází ke ztrátě síly v laně.



Obr. 4.10: Simulace platformy při síle 5 N



Obr. 4.11: Chyba průběhu souřadnic simulace platformy při síle 5 N



Obr. 4.12: Průběh sil při simulaci 5 N platformy

Závěr

Diplomová práce se zabývala možností automatického sestavování pohybových rovnic soustavy těles s lany. V první kapitole byla stručně popsána historie vývoje dynamických modelů tensegritických struktur od R. Skeltona. Byl vysvětlen důvod volby fyzikálních souřadnic spolu s využitím Newton-Eulerových rovnic místo využití prostředí SIMSCAPE MBS.

Ve druhé kapitole byly odvozeny pohybové rovnice pro soustavu těles spojených sférickými vazbami, lany, pružinami a tlumiči. Spolu s odvozenými rovnicemi byly popsány použité symboly a struktura vektorů popisující dynamiku těles. V sekci 2.7 byla diskutována problematika stability řešení soustavy DAE rovnic. První dvě kapitoly splnily první cíl práce.

Struktura vytvořeného softwaru byla popsána ve třetí kapitole. Byly popsány vstupní soubory softwaru. Procesy stojící za jednotlivými metodami, které jsou součástí vytvořeného softwaru, byly v této kapitole popsány za použití diagramů aktivit. Byla diskutována volba řešice ODE15s pro modelované mechanické úlohy. Popsáním struktury a vytvořením softwaru byl splněn druhý cíl práce.

Výsledky dynamické simulace vytvořeného softwaru byly porovnány s modely vzniklými v prostředí SIMSCAPE MBS. Byly ukázány výsledky dvou testovacích úloh. První úlohou byla modifikovaná úloha dvojkyvadla. Druhým příkladem byla struktura tvořená třemi tyčemi a jednou deskou držící pohromadě pomocí lan. Porovnání výsledků z vytvořeného softwaru s prostředím SIMSCAPE ukázalo velkou míru shody obr. 4.4, obr. 4.9 a obr. 4.11.

Vytvořený software je připravený pro použití při modelování struktur složených z tuhých těles spojených sférickou vazbou, lany, pružinami nebo tlumiči. Software byl vytvořen s ohledem na možnost počítání inverzní dynamické úlohy, která ale nebyla cílem této práce.

Dynamickou simulací testovacích struktur s porovnáním s modelem z prostředí MATLAB-SIMULINK-SIMSPACE MBS byl splněn i třetí cíl této práce a to verifikace softwaru. Všechny cíle práce byly splněny.

Seznam použitých zdrojů

- [1] R. Motro. *Tensegrity: Structural Systems for the Future*. Elsevier Science, 2003. ISBN 9780080542348.
- [2] Muhao Chen, Jun Chen, Manoranjan Majji, and Robert E. Skelton. Tensegrity system dynamics in fluids, nonlinear dynamics. *Springer Verlag*, 2023. In press. hal-03287826.
- [3] M. C. Skelton, R. E. a Oliveira. *Tensegrity Systems*. Springer, Boston, 2009. ISBN 978-0-387-74241-0.
- [4] John Rieffel and Jean-Baptiste Mouret. Adaptive and resilient soft tensegrity robots. *Soft Robotics*, pages 318–329, Jun 2018. doi: 10.1089/soro.2017.0066. URL <http://doi.org/10.1089/soro.2017.0066>.
- [5] Haresh Karnan, Raman Goyal, Manoranjan Majji, Robert E. Skelton, and Purneet Singla. Visual feedback control of tensegrity robotic systems. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2048–2053, 2017. doi: 10.1109/IROS.2017.8206022.
- [6] Jing Yao Zhang and Makoto Ohsaki. *Tensegrity Structures*. Mathematics for Industry. Springer Tokyo, 1 edition, 2015. doi: 10.1007/978-4-431-54813-3.
- [7] Raman Goyal, Muhao Chen, Manoranjan Majji, and Robert E. Skelton. Notes: Modeling of tensegrity structures. *Journal of Open Source Software*, 4(42): 1613, 2019. doi: 10.21105/joss.01613. URL <https://doi.org/10.21105/joss.01613>.
- [8] Martin Prokop. Automatická tvorba modelu tensegritické struktury v prostředí matlab simulink a Simscape. Bakalářská práce, České vysoké učení technické v Praze, 2021.
- [9] Martin Hrabačka. Design and dynamic analysis of active tensegrity structures. Diplomová práce, Pilsen, 2022.
- [10] K. Nagase, T. Yamashita, and N. Kawabata. On a connectivity matrix formula for tensegrity prism plates. *Mechanics Research Communications*, 77: 29–43, 2016. ISSN 0093-6413. doi: <https://doi.org/10.1016/j.mechrescom>.

2016.08.003. URL <https://www.sciencedirect.com/science/article/pii/S0093641316300933>.

- [11] R. Motro, S. Najari, and P. Jouanna. Static and dynamic analysis of tensegrity systems. In C. A. Brebbia and S. A. Orszag, editors, *Shell and Spatial Structures: Computational Aspects*, volume 26 of *Lecture Notes in Engineering*, pages 270–279, 1987.
- [12] Joono Cheong and Robert E. Skelton. Nonminimal dynamics of general class k tensegrity systems. *International Journal of Structural Stability and Dynamics*, 15(02):1450042, 2015. doi: 10.1142/S0219455414500424. URL <https://doi.org/10.1142/S0219455414500424>.
- [13] H. Furuya. Concept of deployable tensegrity structures in space applications. *International Journal of Space Structures*, 7(2):143–151, 1992.
- [14] Cornel Sultan, Martin Corless, and Robert T Skelton. Peak-to-peak control of an adaptive tensegrity space telescope. In *Smart Structures and Materials 1999: Mathematics and Control in Smart Structures*, volume 3667, pages 190–201. SPIE, 1999.
- [15] Cornel Sultan and Robert Skelton. Tensegrity structures prestressability investigation. *International Journal of Space Structures*, 18(1):15–30, 2003.
- [16] Anders Sunde Wroldsen. Modelling and control of tensegrity structures. *Diss. Norwegian University of Science and Technology*, 2007.
- [17] Robert Skelton. Dynamics and control of tensegrity systems. In H. Ulbrich and W. Günthner, editors, *IUTAM Symposium on Vibration Control of Non-linear Mechanisms and Structures*, pages 309–318, Dordrecht, 2005. Springer Netherlands. ISBN 978-1-4020-4161-7.
- [18] Robert Skelton. Efficient dynamic models of tensegrity systems. In *Proc. SPIE 7286, Modeling, Signal Processing, and Control for Smart Structures 2009*, volume 72860P. SPIE, 2009. doi: 10.1117/12.819376.
- [19] Raman Goyal and Robert E Skelton. Tensegrity system dynamics with rigid bars and massive strings. *Multibody System Dynamics*, 46(3):203–228, 2019.
- [20] Raman Goyal, Manoranjan Majji, and Robert E. Skelton. Robust shape control of gyroscopic tensegrity robotic arm. *CoRR*, abs/2011.03829, 2020. URL <https://arxiv.org/abs/2011.03829>.
- [21] Raman Goyal, Muhao Chen, Maoranjan Majji, and Robert E. Skelton. Gyroscopic tensegrity robots. *IEEE Robotics and Automation Letters*, 5(2):1239–1246, 2020. doi: 10.1109/LRA.2020.2967288.

- [22] M. Valášek, V. Stejskal, and J. Březina. *Mechanika A, Skriptum FS ČVUT v Praze*. Vydavatelství ČVUT, Praha, 2002.
- [23] Hubert Hahn. *Rigid Body Dynamics of Mechanisms*. Springer, Berlin, Heidelberg, 1 edition, 2002. doi: 10.1007/978-3-662-04831-3.
- [24] M. Valášek, V. Bauma, and Z. Šika. *Mechanika B, Skriptum FS ČVUT v Praze*. Vydavatelství ČVUT, Praha, 2004.
- [25] M. Stejskal, V. a Valášek. *Kinematics and dynamics of machinery*. Marcel Dekker, New York, 1996. ISBN 08-247-9731-0.

Přílohy

A Rozšířený postup zadávání nové struktury

V této příloze bude popsán postup zadávání vyšetřované struktury krok po kroku. Zadávanou strukturou bude úloha dvojkyvadla popsaná v kapitole 4.

A.1 Stažení softwaru

Software je možné stáhnout v příloze na webu ČVUT DSpace z přílohy k této práci.

A.2 Spuštění softwaru

Pro spuštění softwaru je třeba:

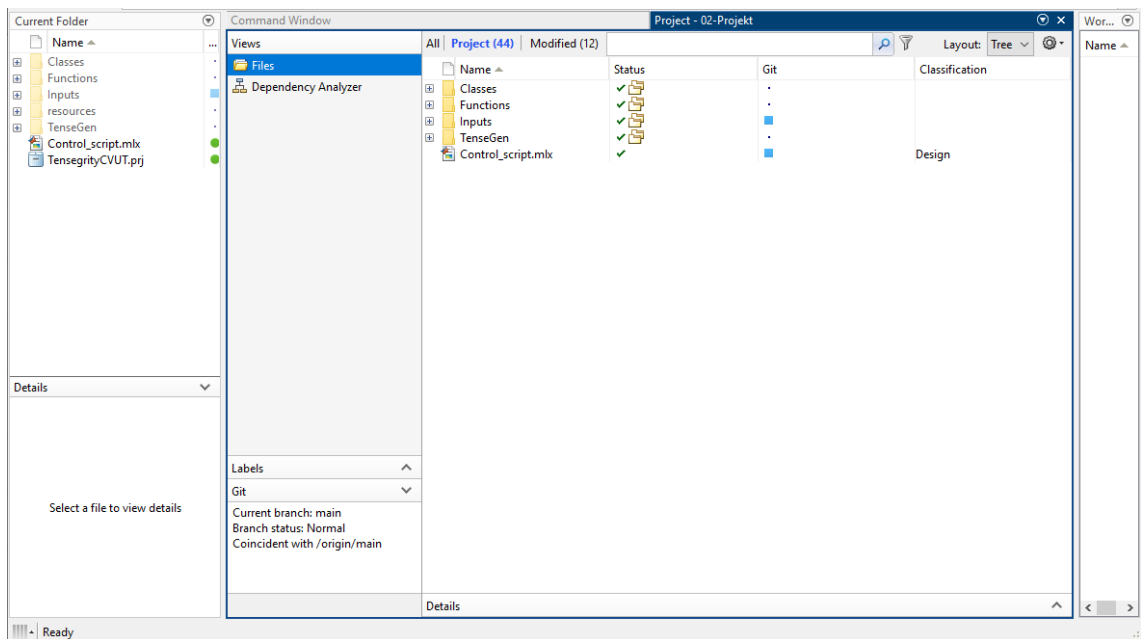
1. Mít stažený MATLAB 2022a a novější
2. MATLAB otevřít
3. Najít složku se softwarem
4. Spustit soubor TensegrityCVUT.prj nacházející se ve složce 02-Projekt

Po spuštění souboru TensegrityCVUT.prj se uživateli otevře prostředí MATLAB projekt obr. 13. Ten načte všechny potřebné cesty k souborům tvořící projekt. Uživatel může spustit soubor Control_script.mlx a vyzkoušet, zda je software stažen a inicializován správně obr. 14.

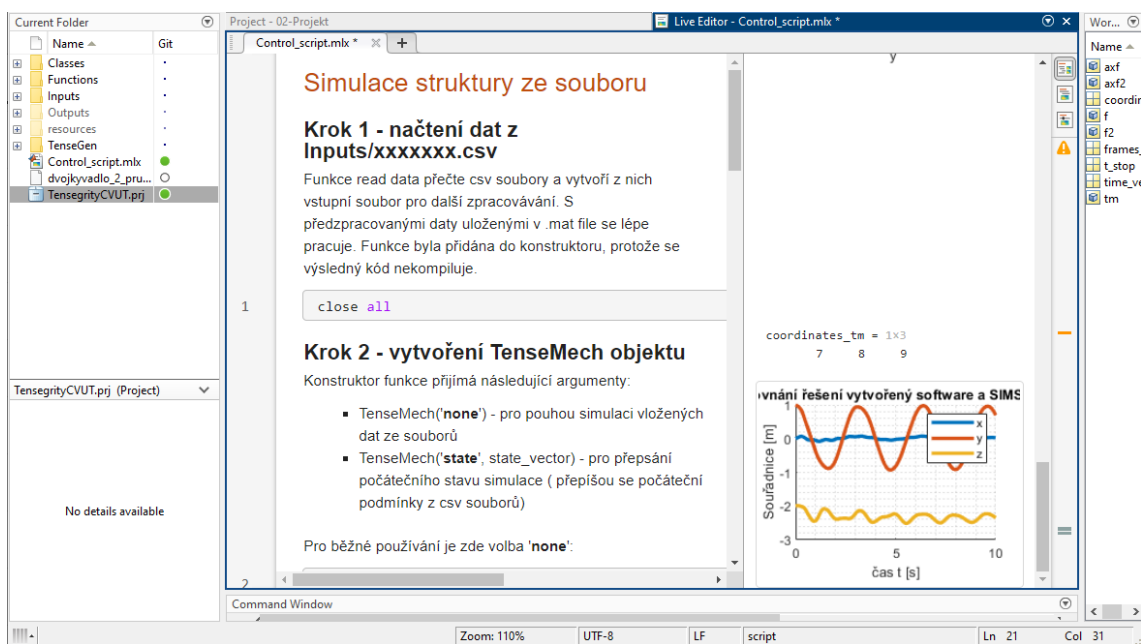
A.3 Zadávání vstupů nové struktury

Pro zadání nové struktury je vhodné si udělat náčrt vyšetřované struktury na papír podobně jako tomu je v schématech 4.1, 4.5 a 4.6. Postup při vytváření nového náčrtu struktury je:

- Zakreslení globálního souřadnicového systému (systém rámu)
- Načrtnutí všech těles, které je za cíl modelovat a očíslování je 1-9



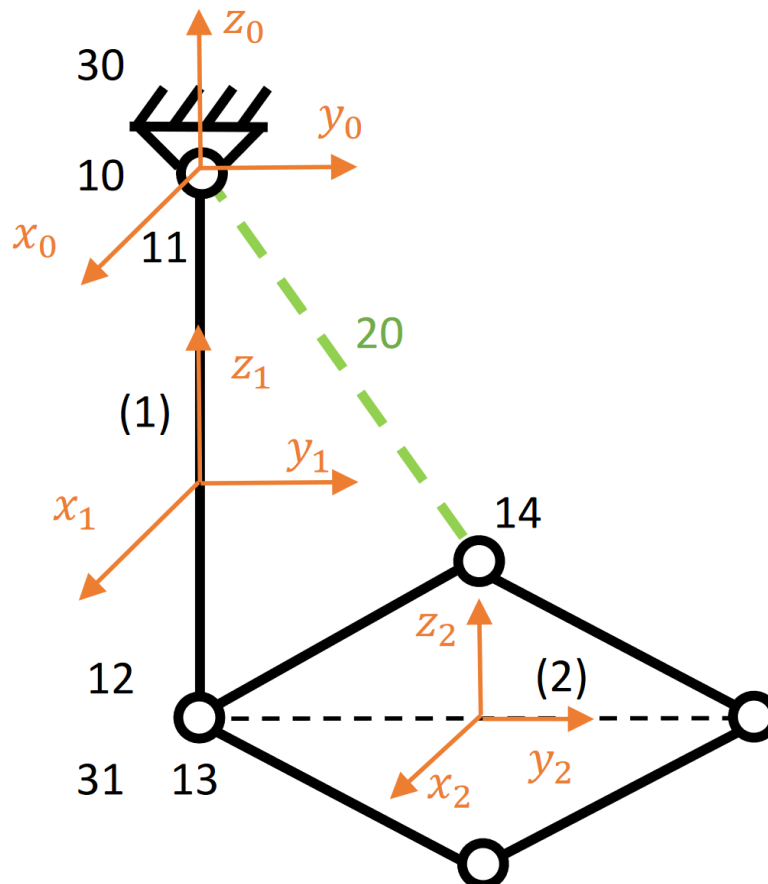
Obr. 13: Inicializovaný MATLAB projekt



Obr. 14: Otevřený soubor Control_script.mlx

- Vyznačení všech lokálních rádiusvektorů ze středů hmotnosti těles a označení je čísly 11-30
 - Pokud je potřeba bod spojit s rámem, bude spojen s tělesem s ID 0
- Zakreslení lokálních souřadnicových systémů každého z těles
- Zapsání souřadnic rádiusvektorů bodů v lokálním souřadnicovém systému
- Zakreslení lan, pružin a tlumičů do schematu a zapsání si které uzly spojují
- Zakreslení kloubových vazeb spojující dva body
- Výpočet počáteční polohy hmotných středů těles vůči souřadnicovému systému rámu
 - Posun vyjádřen v souřadnicovém systému rámu
 - Cardanovy úhly pro popis natočení tělesa vůči rámu

Náčrt pro dvojkyvadlo je zobrazen na obr. 15.



Obr. 15: Náčrt řešeného modelu dvojkyvadla s lanem

Po dokončení náčrtu přichází na řadu vyplnit tabulky vstupů. Do tabulky *BODY.CSV* se zapíše ID těles 0-9, jejich hmotnost, složky tenzoru setrvačnosti lokálním souřadnicovým systémem, počáteční poloha vůči rámu vyjádřená v souřadnicovém systému rámu a Cardanovi úhly pro počáteční vzájemné natočení. Tabulka s vyplněnými údaji je vidět v tab. 9.

ID	Mass	I_x	I_y	I_z	D_{xy}	D_{xz}	D_{yz}	x_0	y_0	z_0	ϕ_{x0}	ϕ_{y0}	ϕ_{z0}	v_{x0}	v_{y0}	v_{z0}	ω_{x0}	ω_{y0}	ω_{z0}
1	30	10	10	0.01	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
2	350	30	30	60	0	0	0	0	1	-2	0	0	0	0	0	0	0	0	0

Tabulka 9: Vstupní data pro úlohu dvojkyvadlo zapsaná v souboru BODY.CSV

Vyplněný soubor BODY.CSV otevřený pomocí programu EXCEL je vidět na obr. 16.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ID	Mass	lx	ly	lz	Dxy	Dxz	Dyz	x0	y0	z0	px0	py0	pz0	vx0	vy0	vz0	dpx0	dpy0	dpz0
2	1	30	10	10	0.01	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
3	2	350	30	30	60	0	0	0	0	1	-2	0	0	0	0	0	0	0	0	0

Obr. 16: Ukázka vyplněného souboru BODY.CSV

Druhý soubor POINT.CSV obsahuje lokální rádiusvektory. Každý řádek reprezentuje jeden lokální rádiusvektor. ID se vyplňují 11-29, IDCON značí ID tělesa, ke kterému je bod pevně připojen (0 pro rám). X,Y,Z pak slouží pro specifikaci rádiusvektoru bodu v souřadnicovém systému tělesa. Vyplněné hodnoty jsou vidět v tab. 10.

ID	IDCON	x	y	z
10	0	0	0	0
11	1	0	0	1
12	1	0	0	-1
13	2	0	-1	0
14	2	-1	0	0
15	2	0	1	0
16	2	1	0	0

Tabulka 10: Vstupní data pro úlohu dvojkyvadlo zapsaná v souboru POINT.CSV

Samotný soubor POINT.CSV je zobrazen na obr. 17.

Zbylé soubory mají už jednoduché vyplnění údajů. U pružin a tlumičů se specifikuje jejich ID, které dva body spojují (IDFROM a IDTO), vyplní se jejich charakteristická veličina (tuhost nebo tlumení) a u pružin navíc jejich volná délka. U lan se vyplňuje jejich ID, měrná tuhost, měrné tlumení, volná délka a posloupnost čísel oddělená čárkami reprezentující ID uzlů, ke kterým jsou připojeny (např. 12,15,18). Data pro lana jsou zobrazeny v tab. 11.

	A	B	C	D	E	F
1	ID	IDCON	IDREF	x	y	z
2	10	0	0	0	0	0
3	11	1	0	0	0	1
4	12	1	0	0	0	-1
5	13	2	0	0	-1	0
6	14	2	0	-1	0	0
7	15	2	0	0	1	0
8	16	2	0	1	0	0

Obr. 17: Ukázka vyplněného souboru POINT.CSV

ID	k	b	l_0	ID_THROUGH
20	10000	100	1.5	11,14

Tabulka 11: Vstupní data pro úlohu dvojkyvadlo zapsaná v souboru CABLE.CSV

Ukázka souboru CABLE.CSV je vidět na obr. 18.

Pro specifikaci kloubové vazby stačí vyplnit její ID a ID dvou bodů, které mají být spojeny. U kloubových vazeb je problém, aby byly zadány počáteční podmínky těles v souladu s vazbovými rovnicemi polohy i rychlosti. Tzn. aby platili rovnice (2.75) a (2.76). Jednoduše řečeno, je třeba specifikovat počáteční podmínky tak, aby uzly spojené sférickou vazbou byly na sobě a měli stejnou rychlost při startu simulace. Data vyplněná pro přidání dvou sférických vazeb jsou zobrazena v tab. 12.

Soubor JOINT.CSV je zobrazen na obr. 19.

A.4 Vykreslení zadané struktury

Po vyplnění vstupních souborů je možné si nechat zadanou strukturu vykreslit. Vytvoření grafické reprezentace zadaných vstupů umožní uživateli jednoduše zjistit, jestli jeho zadaná data odpovídají požadované struktuře.

	A	B	C	D	E
1	ID	k	b	l0	ID_THROUGH
2	20	10000	100	1.5	11,14

Obr. 18: Ukázka vyplněného souboru CABLE.CSV

ID	ID_FROM	ID_TO
50	10	11
51	12	13

Tabulka 12: Vstupní data pro úlohu dvojkyvadlo zapsaná v souboru JOINT.CSV

	A	B	C
1	ID	ID_FROM	ID_TO
2	50	10	11
3	51	12	13

Obr. 19: Ukázka vyplněného souboru JOINT.CSV

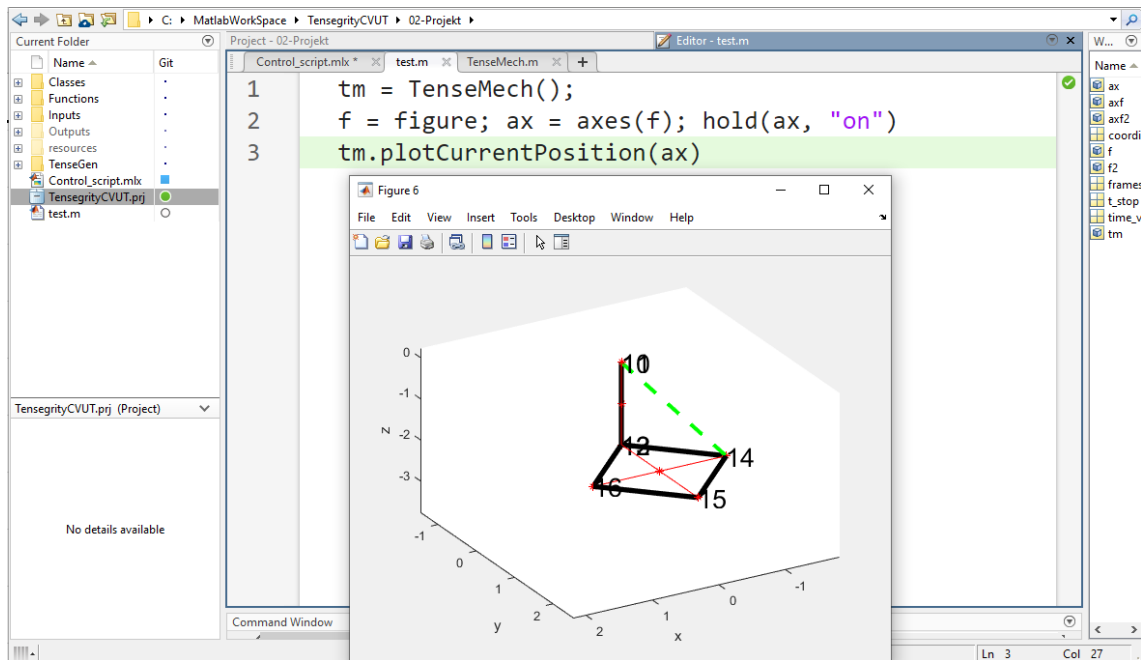
Pro vykreslení struktury je třeba:

1. Mít zapnutý MATLAB projekt ze sekce A.2
2. Vytvořit nový soubor typu .m nebo .mlx a uložit ho do složky 02-Projekt
3. Vytvořit novou instanci objektu TenseMech
4. Vytvořit graf (objekt typu Figure a Axes) do kterého se má struktura vykreslit
5. Zavolat metodu *PlotCurrentPosition(ax)*

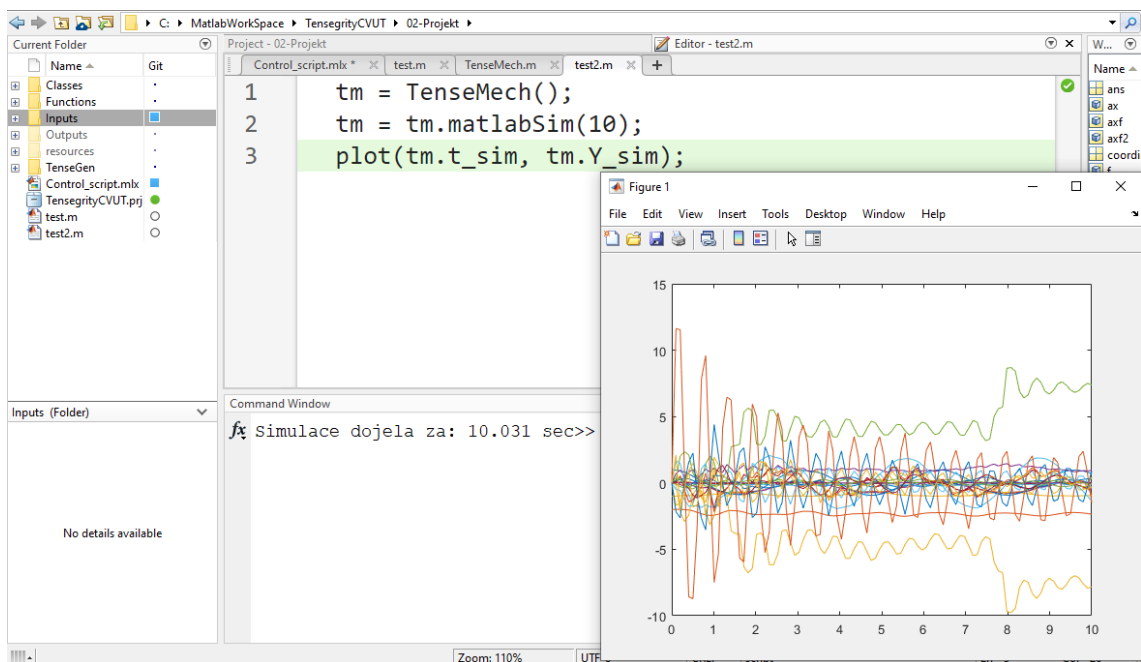
Ukázkový kód je zobrazen na obr. 20 a snímek obrazovky kódu s výstupem je ukázán na obr. 21. Soubor byl pojmenován jako *test.m*. Výsledkem je graf počáteční polohy struktury, odpovídající náčrtu.

```
tm = TenseMech();
f = figure; ax = axes(f); hold(ax, "on")
tm.plotCurrentPosition(ax)
```

Obr. 20: Příkaz pro vykreslení počáteční polohy



Obr. 21: Ukázka souboru pro vykreslení počáteční polohy test.m



Obr. 22: Ukázka souboru pro dynamickou simulaci

A.5 Spuštění simulace

Poté co je uživatel spokojen se zadanými vstupy následuje fáze simulace struktury. Pro čistý vývoj struktury bez žádných vnějších sil stačí spustit kód zobrazený na obr. 23.

```
tm = TenseMech();  
tm = tm.matlabSim(10);
```

Obr. 23: Příkaz pro dynamickou simulaci

Kód vytvoří instanci objektu `TenseMech` a následně se provede simulaci o délce 10 vteřin. Řešení simulace je uloženo ve formě matice obsahující jako řádky stavové vektory \mathbf{x} popsané v sekci 3.2. Matice řešení je uložena v instanci objektu pod názvem `Y_sim`. Pro zobrazení průběhu všech souřadnic je možné použít kód na obr. 24:

```
tm = TenseMech();  
tm = tm.matlabSim(10);  
plot(tm.t_sim, tm.Y_sim);
```

Obr. 24: Příkaz pro dynamickou simulaci s vykreslením souřadnic

Ukázku výsledku kódu lze vidět na obr. 22.

Základním nastavením dynamické simulace je ukládání výsledků ve 100 časových bodech. Pokud by bylo potřeba získat hladší řešení, je možné využít přidání vektoru pro specifikaci časových bodů, ve kterých je žádoucí mít řešení uložené. Např. pokud uživatel chce mít uložených každou 1 ms simulace, lze použít kód na obr. 25.

```
t_stop = 10;  
frames_per_second = 1000;  
time_vector = linspace(0, t_stop, t_stop*frames_per_second);  
tm = TenseMech();  
tm = tm.matlabSim(t_stop, time_vector);  
plot(tm.t_sim, tm.Y_sim);
```

Obr. 25: Příkaz pro dynamickou simulaci s vykreslením souřadnic

V kódu se vytvoří proměnná specifikující kolik vteřin simulace je třeba odsimulovat `t_stop`, kolik časových snímků za vteřinu je třeba uložit specifikuje proměnná `frames_per_second`. Vektor `time_vector` v sobě obsahuje rovnoměrně rozdělené body, ve kterých je vyžadováno řešení.

A.6 Animace struktury

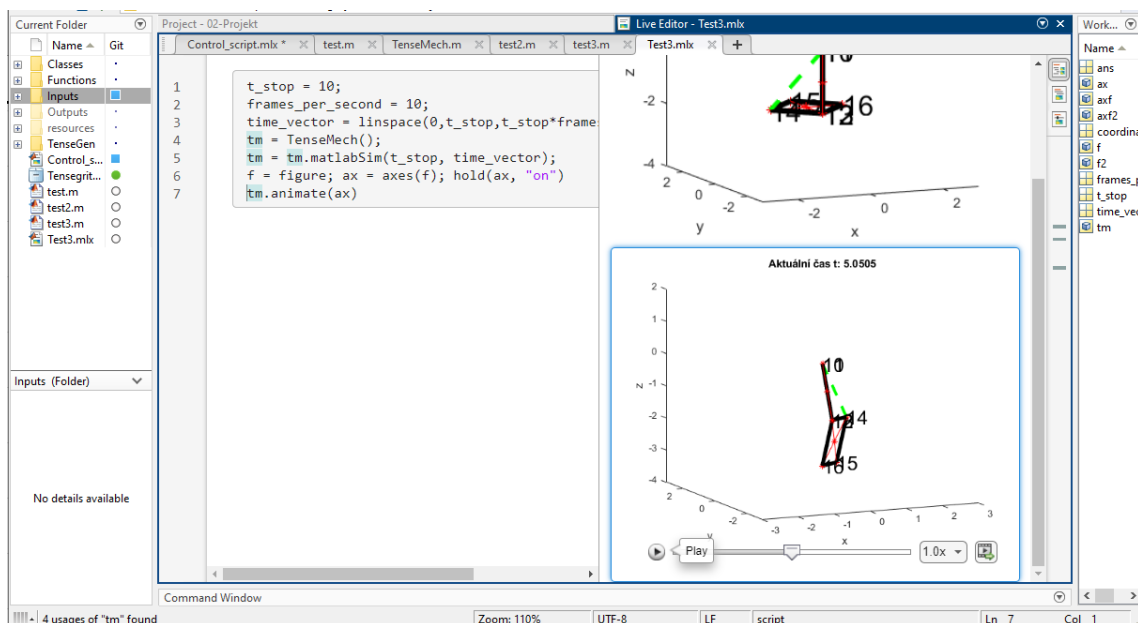
Uživatel má možnost vytvoření animace vypočítaného výsledku. K tomu byla připravena metoda *animate*. Tu lze spustit po dokončení dynamické simulace. Metoda funguje tak, že vždy nastaví stavový vektor popisující konfiguraci těles a spustí metodu *plotCurrentPosition*. Nové stavové vektory jsou získávány postupně z proměnné *Y_sim*. Pokud je vektor řešení *Y_sim* velmi velký (bylo zvoleno mnoho bodů řešení) bude animace probíhat velmi dlouho, protože budou postupně vykreslovány všechny časové okamžiky. Při rozhodnutí spustit animace je žádoucí, aby řešení dynamiky neobsahovalo příliš časových okamžiků.

Pro animaci je potřeba vytvořit nový graf, do kterého bude animace postupně vykreslována. Ukázku kódu pro spuštění animace lze vidět na obr. 26

```
t_stop = 10;
frames_per_second = 10;
time_vector = linspace(0,t_stop,t_stop*frames_per_second);
tm = TenseMech();
tm = tm.matlabSim(t_stop, time_vector);
f = figure; ax = axes(f); hold(ax, "on")
tm.animate(ax)
```

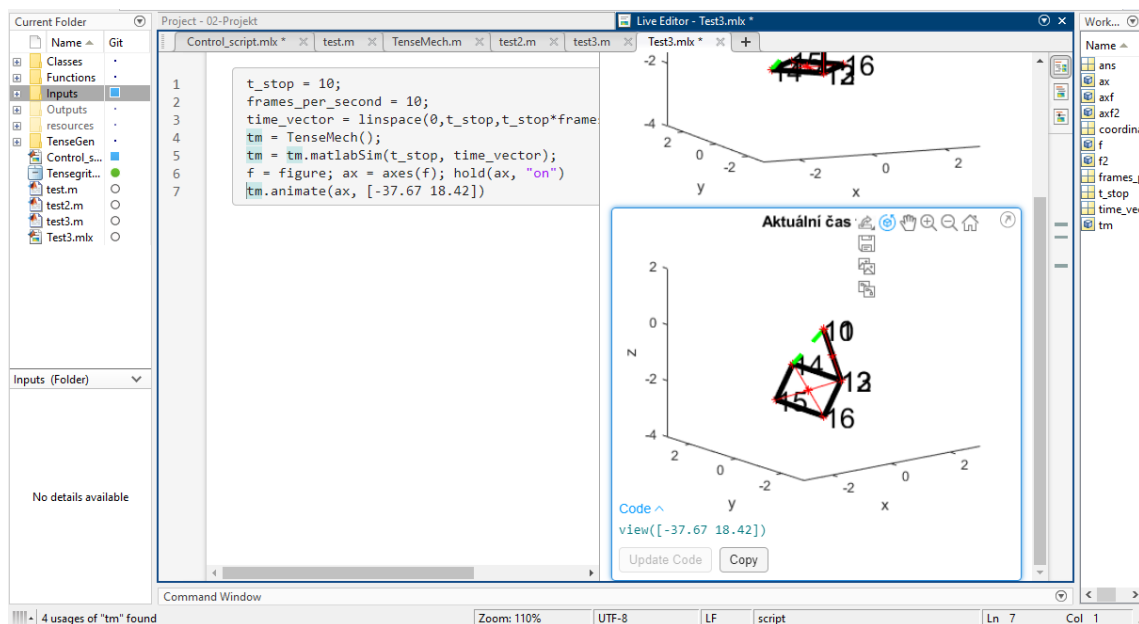
Obr. 26: Ukázka kódu pro spuštění animace

Při práci s animacemi je vhodné použití prostředí live scriptu (přípona *.mlx*). To umožňuje přehrání vytvořené animace vícekrát za sebou bez nutnosti znovuspouštění kódů. Umožňuje i animaci zastavit nebo přetočit dozadu. Ukázku práce s *.mlx* souborem lze vidět na obr. 27.



Obr. 27: Ukázka práce s live scriptem

Občas je potřeba animaci vykreslit z jiného úhlu pohledu. K tomu je připraven druhý arument metody animate. Pro vykreslení animace z jiného úhlu pohledu lze specifikovat vektor pohledu. Ten lze získat pomocí přetočení animace nakonec a použití nástroje rotace. Po jeho použití se objeví dole kód `view` s vektorem o dvou prvcích. Hranatou závorku lze zkopírovat a vložit jako druhý argument metody animate. Ukázku získání view vektoru a jeho následného vložení jako druhého argumentu lze vidět na obr. 28.



Obr. 28: Získání view vektoru

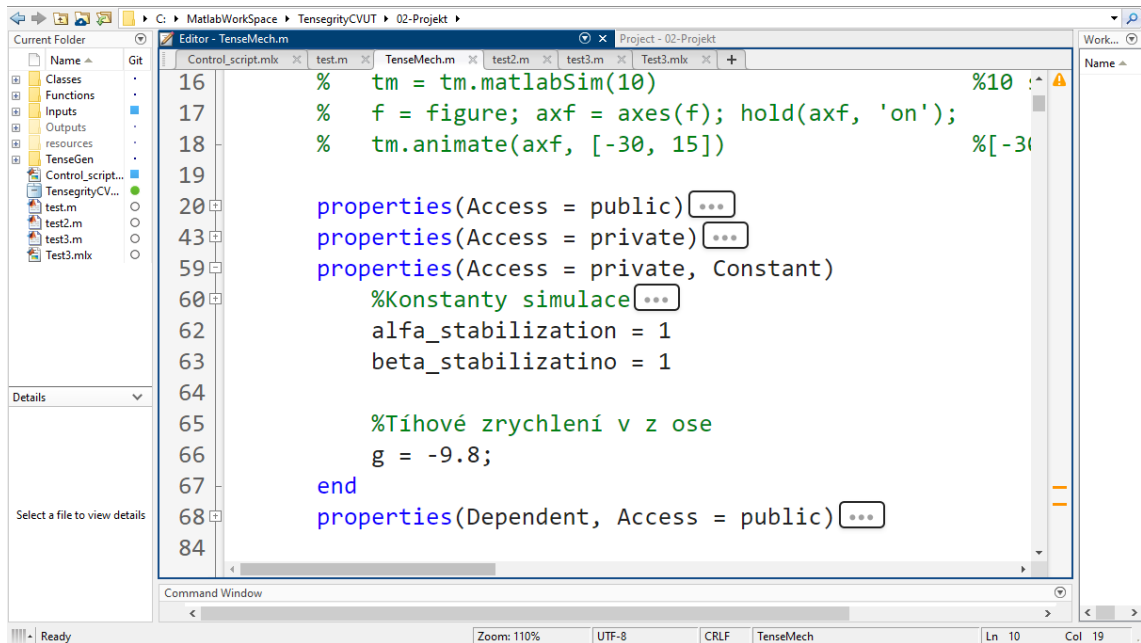
A.7 Úprava konstant simulace

Všechny konstanty jsou specifikovány v souboru 02-Projekt/Classes/TenseMech.m. Ten obsahuje definici třídy TenseMech. Do zde definovaných konstant patří koeficienty stabilizace α a β nebo velikost tíhového zrychlení g . Umístění konstant simulace lze vidět na obr. 29.

A.8 Úprava metody matlabSim

Uživatel má možnost upravit funkci matlabSim, ve které probíhá řešení dynamické úhlohy struktury. Mezi možné změny patří:

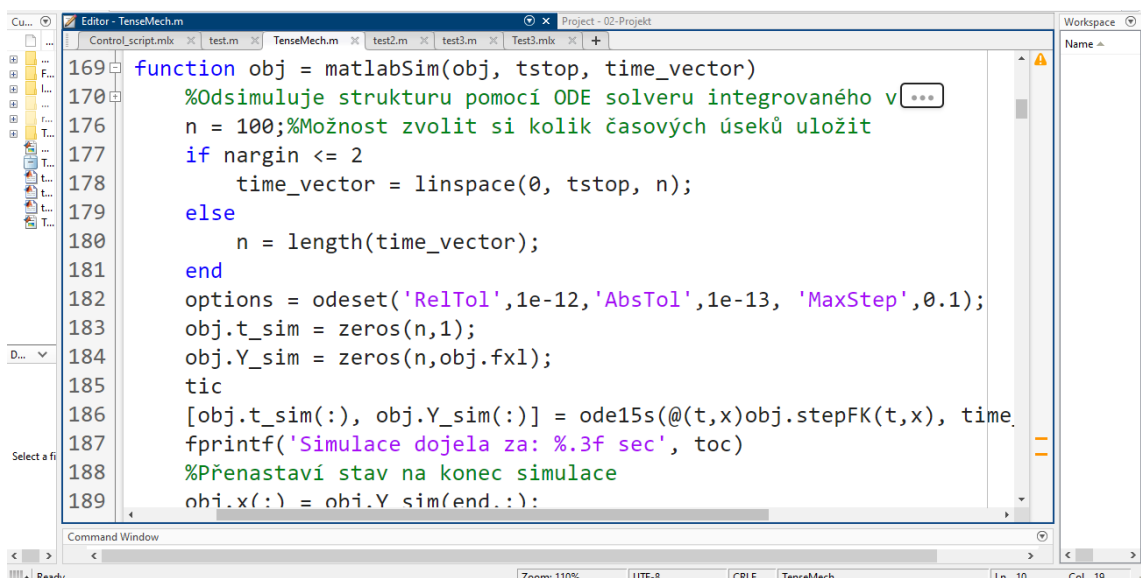
- Změna nastavení řešiče
 - Změna toleranci (přesnosti řešení)
 - Změna maximálního kroku
 - Přidání speciálních vlastností nastavení řešiče



Obr. 29: Umístění konstant

- Změna řešiče z ODE15s na jiný

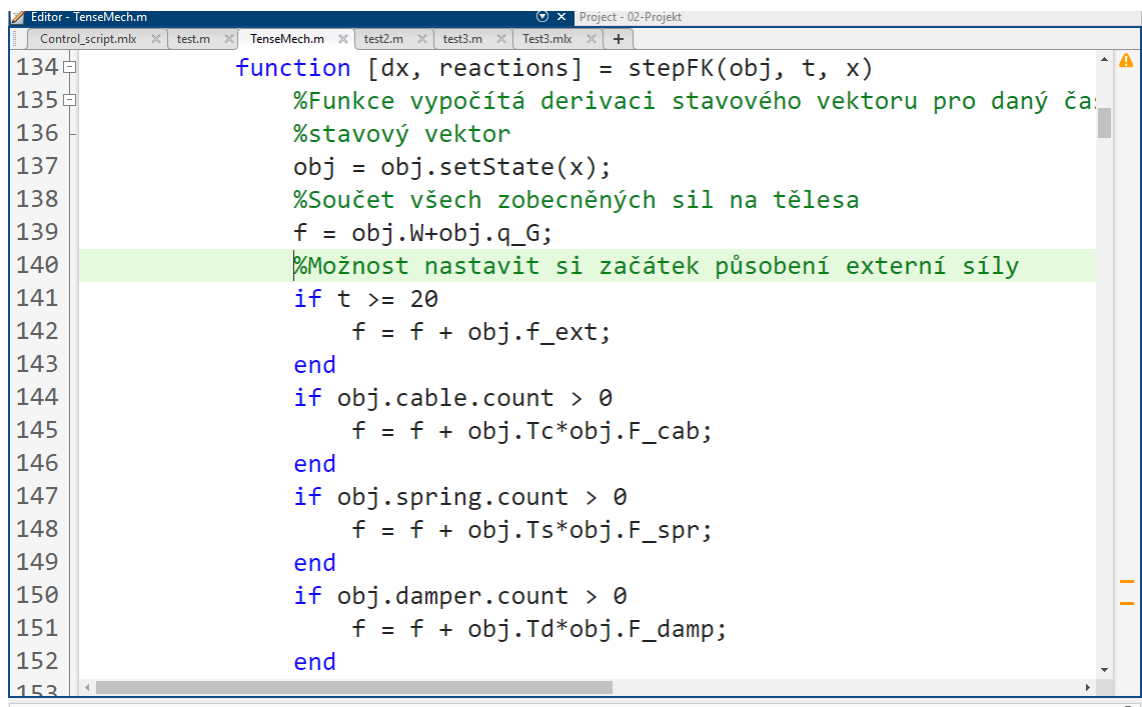
Na obr. 30 lze vidět metodu matlabSim. V první části se nastaví time_vector. Pokud nebyl specifikován vytvoří se rovnoměrně rozdělený vektor o 100 prvcích mezi počátečním a koncovým časem. Následuje vytvoření objektu nastavení, který uživatel může měnit (řádek 182). Hlavní část metody je zavolání numerického řešiče diferenciálních rovnic. Na popisovaném obrázku je použit ODE15s (řádek 186). Řešič volá funkci stepFK, která vrací derivaci stavového vektoru.



Obr. 30: Metoda matlabSim

A.9 Úprava metody stepFK

Metoda stepFK počítá derivaci stavového vektoru. Vstupní argumenty funkce jsou aktuální čas a stavový vektor. Metoda nejprve aktualizuje všechny části instance objektu tak, aby odpovídaly novému stavovému vektoru (řádek 137). Následuje výpočet vnějších sil. Prvně se do vektoru vnějších sil přičtou tíhové síly označené symbolem \mathbf{W} a gyroskopické síly označené jako \mathbf{g}_G (řádek 139). Poté lze nastavit okamžik, kdy mají na strukturu působit vnější síly. Podmínka $t \geq 20$ říká, že pokud je čas větší než 20 vteřin, začne se k vektoru vnějších sil připočítávat vektor vnějších sil \mathbf{f}_{ext} (řádky 141-142). Takto byla zadávána síla v kapitole 4 v úloze platforma. Podmínky pro přidávání externích sil a jejich velikosti lze v této části volit libovolně. Následuje výpočet vnějších sil, do kterých se řadí síly od lan, tlumičů a pružin. To znamená výpočet jejich distribučních matic a velikostí sil. Popsaný proces lze vidět na obr. 31 (řádky 145, 148 a 151).



```
134 function [dx, reactions] = stepFK(obj, t, x)
135     %Funkce vypočítá derivaci stavového vektoru pro daný čas
136     %stavový vektor
137     obj = obj.setState(x);
138     %Součet všech zobecněných sil na tělesa
139     f = obj.W+obj.q_G;
140     %Možnost nastavit si začátek působení externí síly
141     if t >= 20
142         f = f + obj.f_ext;
143     end
144     if obj.cable.count > 0
145         f = f + obj.Tc*obj.F_cab;
146     end
147     if obj.spring.count > 0
148         f = f + obj.Ts*obj.F_spr;
149     end
150     if obj.damper.count > 0
151         f = f + obj.Td*obj.F_damp;
152     end
153
```

Obr. 31: Ukázka funkce metody stepFK část 1

Druhou částí metody stepFK je výpočet zbylých matic (obr. 32). Všechny z těchto matic jsou počítány po jejich zavolání díky využití vlastnosti *Dependent*. Ta umožňuje definovat atribut objektu s tím, že atribut je vypočítán po zavolání atributu. Díky tomu lze přehledně sestavit matici systému \mathbf{A} , vektor pravé strany \mathbf{B} a nalézt řešení v daném časovém okamžiku (v kódu označeno jako *res*). Řešení obsahuje derivaci stavového vektoru a vektor Lagrangeovo multiplikátorů. Tyto dva vektory je třeba od sebe odseparovat. Matice \mathbf{A} a vektor \mathbf{B} vychází z rovnice 2.128.

```

150     if obj.damper.count > 0
151         f = f + obj.Td*obj.F_damp;
152     end
153     %Jakobiho matice
154     Phi_c = obj.Phi;
155     %Samotné řešení dynamiky
156     A = [eye(obj.hxl),          zeros(obj.hxl), zeros(obj.hxl, obj.nc)
157         zeros(obj.hxl),        obj.M,          Phi_c'
158         zeros(obj.nc, obj.hxl), Phi_c,          zeros(obj.nc)];
159     B = [obj.T*obj.x((obj.hxl+1):end);
160         f;
161         obj.beta_c-obj.beta_stabilizacino*Phi_c*obj.x((obj.hxl+1):end);
162     res = A\B;
163     %Separace derivace stavů a reakcí
164     dx = res(1:obj.fx1);
165     reactions = res((obj.fx1+1):end);
166 end
167
168 %Metoda Simulace v Matlabu

```

Obr. 32: Ukázka funkce metody stepFK část 2

A.10 Výpočet inverzní úlohy

Vytvořený software byl navržen s ohledem na možnost počítání inverzní dynamické úlohy. Je tedy možné vytvořit novou metodu, nazvěme ji *stepID*, která bude mít jako vstup požadovaný vektor zrychlení souřadnic a jako výstup potřebné síly v lanech a Lagrangeovy multiplikátory pro dosažení požadovaného zrychlení.

Pro odvození rovnice pro výpočet potřebných sil v lanech se vyjde z rovnice (2.123) do které se dosadí distribuční matice z rovnice (2.130). Při tomto odvození budou uvažována pouze lana (tedy systém bez pružin a tlumičů). Dále se z rovnice (2.123) vyjme první řádek přepočítávající vektor rychlostí na vektor derivace souřadnic. Výsledek je vidět v rovnici (1).

$$\begin{bmatrix} \mathbf{M}(\mathbf{s}) & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \\ \mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s}) & \mathbf{0}_{n_c \times n_c} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_c\mathbf{F}_c + \mathbf{W} + \mathbf{q}_G \\ \boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v}) \end{bmatrix} \quad (1)$$

Dalším krokem odvození je roztrnutí maticové rovnice (1) na dvě samostatné rovnice:

$$\mathbf{M}(\mathbf{s})\dot{\mathbf{v}} - \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s})\boldsymbol{\lambda} = \mathbf{T}_c\mathbf{F}_c + \mathbf{W} + \mathbf{q}_G \quad (2)$$

$$\mathbf{g}_s(\mathbf{s})\mathbf{T}(\mathbf{s})\dot{\mathbf{v}} = \boldsymbol{\beta}_c(\mathbf{s}, \mathbf{v}) \quad (3)$$

Rovnici (2) je dále možné upravit do podoby:

$$\mathbf{T}_c\mathbf{F}_c + \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s})\boldsymbol{\lambda} = \mathbf{M}(\mathbf{s})\dot{\mathbf{v}} - \mathbf{W} - \mathbf{q}_G \quad (4)$$

A následně lze rovnici (4) převést do maticového tvaru:

$$\begin{bmatrix} \mathbf{T}_c & \mathbf{T}^T(\mathbf{s})\mathbf{g}_s^T(\mathbf{s}) \end{bmatrix} \begin{bmatrix} \mathbf{F}_c \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{M}(\mathbf{s})\dot{\mathbf{v}} - \mathbf{W} - \mathbf{q}_G \quad (5)$$

Rovnice (5) popisuje způsob jak vypočítat potřebné síly v lanech pro dosažení požadovaného zrychlení $\dot{\mathbf{v}}$.

Výsledná možná podoba implementace rovnice (5) do třídy TenseMech je zobrazena na obr. 33

```
function [Fc, lambda] = stepID(obj, t, x, dv)
    obj = obj.setState(x);
    A = [obj.Tc, obj.Phi'];
    B = obj.M*dv - obj.W - obj.q_G;
    res = A \ B;
    Fc = res(1:obj.cable.count);
    lambda = res((obj.cable.count+1):end);
end
```

Obr. 33: Ukázka kódu pro výpočet inverzní úlohy

Takto řešený problém předpokládá několik předpokladů. První předpoklad je, že všechna lana jsou řízena. Dále se předpokládá, že je více řízených lan než stupňů volnosti systému. A na závěr se předpokládá, že je dopředu známý vektor zrychlení všech částí systému.

B Vytvořený software

Vytvořený software je dostupný v příloženém souboru typu ZIP.