VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

MODELOVÁNÍ STROJŮ V MATLAB/SIMULINK A SIMSCAPE

MODELING OF MACHINERY IN MATLAB/SIMULINK AND SIMSCAPE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Lukáš Moravanský
AUTHOR

VEDOUCÍ PRÁCE — Ing. et Ing. Stanislav Lang, Ph.D.
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání diplomové práce

Ústav:	Ústav automatizace a informatiky
Student:	**Bc. Lukáš Moravanský**
Studijní program:	Aplikovaná informatika a řízení
Studijní obor:	bez specializace
Vedoucí práce:	**Ing. et Ing. Stanislav Lang, Ph.D.**
Akademický rok:	2023/24

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Modelování strojů v Matlab/Simulink a Simscape

Stručná charakteristika problematiky úkolu:

Práce je zaměřena na pokročilé modelování strojních zařízení s využitím nástroje Matlab/Simulink. Student se v rámci práce seznámí hlouběji s nástroji Matlab/Simulink a nastuduje postupy tvorby modelů s využitím nadstavby Simscape. V rámci své práce bude prezentovat postupy tvorby modelu (nejlépe na případu soustavy tiskařských válců).
Student se v rámci své diplomové práce zapojí do řešení jednoho z projektů řešeného na ústavu v rámci smluvního výzkumu. Výsledky pak využije i ve své práci a bude je prezentovat vhodnou formou tak, aby práce měla pokud možno edukační přínos a prezentovala využití nástrojů na reálném příkladu, zároveň však nesmí být porušena práva třetích stran (součásti podléhající utajení budou vhodně modifikovány či nahrazeny).

Cíle diplomové práce:

Proveďte stručnou rešerši v oblasti fyzikálního modelování strojů.
Nastudujte postupy fyzikálního modelování strojních částí v prostředí Matlab/Simulink (zejména s využitím knihovny Simscape/Driveline).
Popište základní postupy tvorby modelu stroje s využitím Simscape.
Realizujte model stroje či strojní části (ideálně soustavy válců řešené v přidruženém projektu ústavu).
Verifikujte model porovnáním s fyzickým zařízením.
Proveďte zhodnocení výsledků i samotného modelovacího nástroje.

Seznam doporučené literatury:

GREPL, Robert. Modelování mechatronických systémů v Matlab SimMechanics. Praha: BEN, 2007. ISBN 978-80-7300-226-8.

MARIA, Anu. Introduction to modeling and simulation. Proceedings of the 29th conference on Winter simulation - WSC '97 [online]. New York, New York, USA: ACM Press, 1997, 7-13 [cit.2023-10-19]. ISBN 078034278X. Dostupné z: doi:10.1145/268437.268440

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2023/24

V Brně, dne

L. S.

Ing. Pavel Heriban, Ph.D. ředitel ústavu	doc. Ing. Jiří Hlinka, Ph.D. děkan fakulty

ABSTRAKT

Diplomová práce se zaměřuje na vývoj digitálního dvojčete osy formového válce flexotiskového stroje pro firmu SOMA pomocí softwaru MATLAB/Simscape. Teoretické základy a metody soustředěných parametrů byly využity pro tvorbu a validaci fyzikálního modelu, přičemž hlavní důraz byl kladen na přesnost shody s reálným systémem v rámci rezonancí a anirezonancí. Bylo vytvořeno uživatelské rozhraní (GUI) umožňující snadné úpravy a testování modelu. Práce ukazuje výhody propojení nástrojů MATLABu pro vývoj fyzikálních modelů a poskytuje základ pro budoucí úpravy a rozšíření modelu.

ABSTRACT

The thesis focuses on the development of a digital twin of the axis of the mould cylinder of a flexographic printing machine for SOMA using MATLAB/Simscape software. Theoretical foundations and concentrated parameter methods were used to develop and validate the physical model, with the main focus on the accuracy of the match with the real system in terms of resonances and aniresonances. A graphical user interface (GUI) was developed to allow easy modification and testing of the model. The work demonstrates the benefits of linking MATLAB tools for developing physics models and provides a basis for future model modifications and extensions.

KLÍČOVÁ SLOVA

digitální dvojče, MATLAB, Simscape, model se soustředěnými parametry, modelování mechanické soustavy

KEYWORDS

digital twin, MATLAB, Simscape, lumped parameter model, mechanical system modelling

ÚSTAV AUTOMATIZACE A INFORMATIKY

2024

BIBLIOGRAFICKÁ CITACE

MORAVANSKÝ, Lukáš. *Modelování strojů v Matlab/Simulink a Simscape.* Brno, 2024. Dostupné také z: https://www.vut.cz/studenti/zav-prace/detail/157760. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Stanislav Lang.

PODĚKOVÁNÍ

„Rád bych poděkoval Lukáši Skalickému za vedení vývoje projektu a Jiřímu Junkovi za technický dozor. Velké poděkování patří Stanislavu Langovi za zprostředkování projektu na ústavu a za jeho odborné vedení při vypracovávání této práce. Také bych chtěl vyjádřit svou vděčnost mé rodině a mé nastávající ženě za jejich podporu a trpělivost během celé doby mého studia.“

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že, že tato práce je mým původním dílem, vypracoval jsem ji samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury.

Jako autor uvedené práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., včetně možných trestně právních důsledků.

V Brně dne 20. 5. 2024 ……

Lukáš Moravanský

OBSAH

1 ÚVOD

Vývoj nových strojů je složitý a nákladný proces, který vyžaduje důkladné plánování a testování. Fyzické prototypování může být časově a finančně náročné, často vyžaduje opakované úpravy a testování prototypů, které zpomalují celý proces vývoje. Spotřeba materiálů a energie při výrobě fyzických prototypů a s tím související doprava má negativní ekologické dopady.

V oblasti návrhu nového produktu ale i po celou délku jeho životního cyklu přináší digitální dvojče. Během návrhu umožňuje simulovat různé varianty produktu ve virtuálním prostředí a po uvedení na trh poskytuje digitální dvojče možnost provádět prediktivní údržbu. Co je digitální dvojče a kdy se jedná pouze o model?

Pojem *digitální dvojče* lze chápat jako digitální reprezentaci fyzického produktu. V odborné literatuře je možné nalézt různé definice digitálního dvojčete, které se liší v závislosti na konkrétní oblasti, v níž je tento termín používán. Všechny ale vychází z prezentace představené v roce 2002 Michaelem Grivesem “Conceptual Ideal for PLM (Product Lifetime Management)" na Michiganské univerzitě. V závislosti na toku informací mezi fyzickým a digitálním objektem pak byly zavedeny další pojmy jako "Digital Master" nebo "Digital Shadow". Sám Michaele Grivese definuje další typy digitálních dvojčat jako "Digital Twin Prototype" nebo "Digital Twin Instance"[1; 2].

V literatuře je definice *digitálního dvojčete* často vztahována k produktu, kterým může být i například výrobní nebo distribuční proces (často vztahováno ke kybernetico-fyzickým systémům). V takovém případě by při modelování bylo použito jiných přístupů, než kterými se zabývá tato práce. Je tedy vhodné uvést tuto definici do kontextu fyzikálního modelování stroje nebo strojní součásti. *Digitální dvojče* stroje by mělo umožňovat simulaci virtuálního modelu fyzického stroje, který je založen na datech vztahujících se k reálnému stroji a umožňuje dynamickou aktualizaci nebo úpravu modelu [3; 4; 5].

Digitální dvojče nemusí být nutně řízeno daty, ale výsledky simulace by měli být ekvivalentní naměřeným veličinám z fyzického stroje. Jako vstupní parametry modelu budou pravděpodobně použita naměřená data, jako jsou vlastnosti materiálu, zatížení nebo počáteční podmínky [4; 6].

Klíčovým aspektem digitálních dvojčat je tedy jejich propojení s reálnými objekty. Článek [4] uvádí, že digitální dvojče bez fyzikálního dvojčete je pouze model. Vychází při tom z analogie k biologickým dvojčatům.

Tato práce se zabývá fyzikálním modelováním strojů se zaměřením na modelovací nástroj MATLAB a poslední kapitola je věnována projektu digitálního dvojčete části flexotiskového stroje. Teoretická část se věnuje aspektům, ze kterých se při vývoji digitálního dvojčete vycházelo.

Celá diplomová práce do určité míry vychází z poznatků z bakalářské práce [7], která je zaměřena na modelování strojů v nástroji MapleSim a praktická část se také věnuje modelování digitálního dvojčete části flexostiskového stroje. V teoretické části

této práce tedy nejsou zmiňovány základní podklady zabývající se modelováním (multidomain modeling, objektově orientované modelování atd.), protože jsou již popsány v bakalářské práci [7].

Při vývoji modelu v posední kapitole této práce se předpokládalo, že model bude využíván firmou SOMA při návrhu nových konstrukcí barevníků. Proto práce klade důraz na požadavky firmy a validitu modelu.

2 MODELOVÁNÍ MECHANICKÉ SOUSTAVY

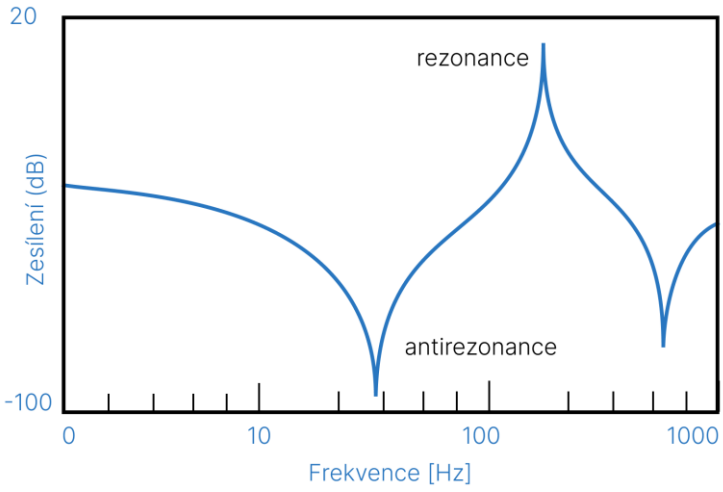
Tato kapitola je zaměřena na teoretické aspekty, ze kterých se vycházelo při vývoji, verifikaci a validaci modelu části flexotiskového stroje popisovaného v kapitole 5. Pozornost je zaměřena na modelování a analýzu torzních vibrací rotační mechanické soustavy. Vibrace mají významný vliv na správnou funkci a životnost stroje. V rámci projektu, který je předmětem pozdější kapitoly, vibrace ovlivňují kvalitu tisku a model mechanické soustavy pomáhá určit vlastní frekvence systému již ve fázi návrhu.

2.1 Rezonance a antirezonance

Každý mechanický systém má svou vlastní frekvenci, při které systém přirozeně osciluje, aniž by na něj působila vnější síla. Tyto frekvence jsou určeny geometrií a materiálovými vlastnostmi daného systému – konkrétně hmotností a tuhostí.

Pokud je mechanický systém buzen a budící frekvence je rovna vlastní frekvenci systému, bude dynamické zesílení nekonečné. V reálném systému není zesílení nekonečně velké, protože každý reálný systém má konečné tlumení (obr. 1). Tento stav se nazývá rezonance a je pro systém rizikový. I přesto, že zesílení není nekonečně velké, dochází v takovém stavu v systému k velkým zesílením způsobující cyklická namáhaní, která mohou vést k únavě materiálu [8].

Vlastní frekvence se ve frekvenční analýze na logaritmické stupnici vyznačují jako „vrcholy“ křivky. Naopak „údolí“ na křivce značí antirezonance. Pokud se budící frekvence blíží k antirezonanční, jsou vibrace v systému tlumeny [9].



Obrázek 1 Rezonance a antirezonance ve frekvenčním spektru

Rezonanční body jsou určeny vibrační analýzou. Vibrodiagnostiku lze provést již na hotovém stroji a úpravou geometrie či materiálu lze změnit rezonanční vlastnosti soustavy. Iterativní úpravy ale mohou znamenat časovou a finanční zátěž, proto je vhodné

vyvinout model soustavy, který lze jednoduše upravovat a provádět na něm vibrační analýzu. Frekvenční analýza z reálného stroje pak poslouží pro validaci modelu.

2.2 Lumped parameter model

Popisované modelovací přístupy a nástroje využívají modelování se soustředěnými prvky. Tento přístup lze ilustrovat na příkladu rezistoru v elektronickém systému. Reálný fyzikální rezistor je tvořen biliony atomů a komplexní strukturou mezi nimi. Namísto modelování obrovského množství interakcí mezi jednotlivými atomy, je rezistor zjednodušen a popsán jednou lineární rovnicí – Ohmovým zákonem [10].

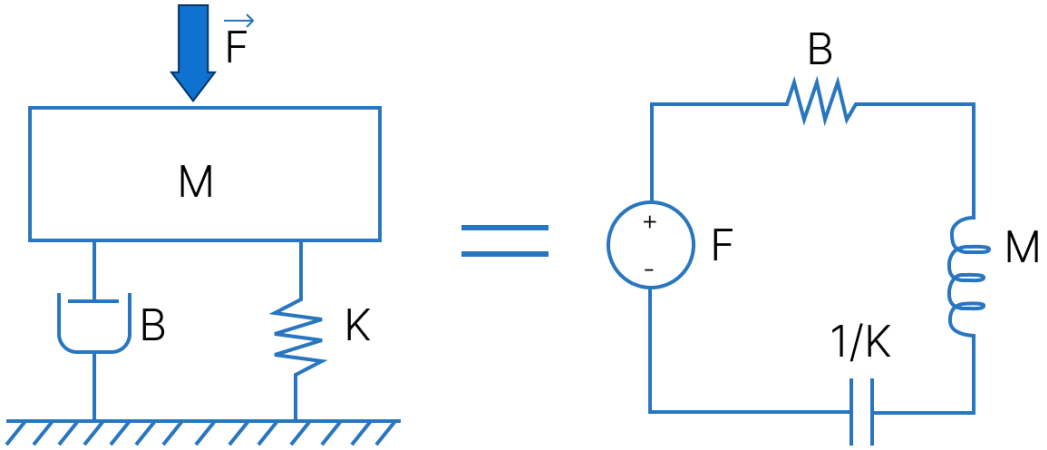
U modelování prvků mechanických soustav se využívá tuhých – nedeformovatelných těles definovaných momentem setrvačnosti, pružin definovaných tuhostí a tlumičů definovaných tlumením. Parametry, kterými jsou prvky definovány, jsou soustředěny do jednoho bodu. U rotační soustavy popisované v kapitole 2.3 tento bod rotuje kolem své osy [11].

U tématu modelování se soustředěnými parametry lze ukázat analogii mezi mechanickými a elektronickými systémy. Zmíněná analogie umožňovala simulovat mechanické soustavy na analogových počítačích, kde mechanické prvky byly reprezentovány pomocí elektronických součástek. Například v elektronickém obvodu je kondenzátor prvek uchovávající elektrickou energii. Stejně tak pružina je prvek, který ukládá mechanickou energii při deformaci. V této analogii představuje elektrický proud v mechanickém systému rychlost. Na obrázku 2 je vyobrazen RLC elektrický obvod ekvivalentní systému „hmotnost-pružina-tlumič“ (mass-spring-damper) [12; 10].

Ze způsobu modelování se soustředěnými prvky vychází přístup modelování pomocí blokových diagramů, který je více popsán v pozdější kapitole 2.5. Zároveň nástroje jako Simulink nebo MapleSim fungují na tomto způsobu modelování fyzikálních systémů.

Zjednodušování prvků systému přináší svá omezení v jeho validitě. Komplexní systémy lze rozdělit do více jednodušších částí modelovaných tímto přístupem. S vyššími požadavky na přesnost modelu je pak možné uvažovat o jiném přístupu modelování, například modelování s distribuovanými parametry. V takovém případě ale vzroste výpočetní náročnost modelu.

U modelu s distribuovanými parametry je hmota objektu rozprostřena do série nekonečně malých hmotnostních elementů. Systém je popisován parciálními diferenciálními rovnicemi. Většinu složitějších systémů nelze řešit analyticky, proto se u řešení modelů používá metoda konečných prvků. Vibrace jsou pak analyzovány pomocí modální analýzy [13].

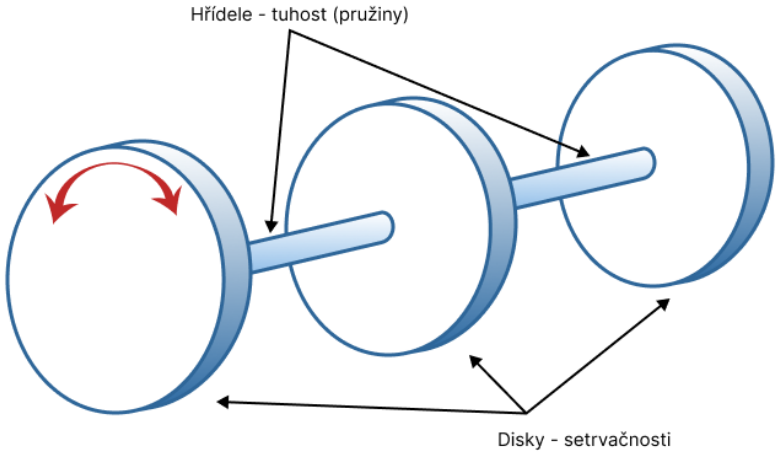


Obrázek 2 Analogie elektronického systému k mechanickému [14]

2.3 Diskový model

Diskový model je běžnou aproximací pro analýzu torzních vibrací v rotačních mechanických systémech. V modelu jsou významné prvky systému zjednodušeny na disky – tuhá tělesa, která jsou charakterizována momentem setrvačnosti. Disky jsou mezi sebou propojeny torzními pružinami, které reprezentují torzní tuhost rotačních součástí. Následně je jeden z disků rozkmitán sinusoidálně se měnícím kroutícím momentem, který způsobí, že všechny prvky začnou vibrovat okolo rovnovážné polohy. Celý tento zjednodušený systém je pak modelován jako *lumped parameter model* (viz kapitola 2.2). Systém je znázorněn na obrázku 3 [8].

V literatuře je tato analýza popisována jako netlumená vzhledem k tomu, že jsou z ní vynechány tlumící prvky. Chyba při výpočtu vlastních frekvencí vzniklá zanedbáním tlumení je prakticky zanedbatelná [8].



Obrázek 3 Lumped Parameter Model podle [8]

Za disky se v literatuře považují prvky systému s momentem setrvačnosti významným pro torzní analýzu. U prvků, jako jsou elektromotory nebo spojky, mohou být údaje o momentu setrvačnosti získány od dodavatelů[8]. V jiném případě lze moment setrvačnosti získat pomocí CAD softwaru, případně po určité aproximaci spočítat jako:

$$I = \frac{\rho \cdot \pi}{32} \cdot (D^4 - d^4) \cdot L \tag{1}$$

kde:

I	=	Moment setrvačnosti	[kg·m^2]
ρ	=	Hustota	[kg·m^{-3}]
D	=	Vnější průměr	[m]
d	=	Vnitřní průměr	[m]
L	=	Délka prvku	[m]

Na uvedeném příkladu je hřídel modelována jako pružina. Ačkoli je aproximace pro model popisovaný v kapitole 5 nedostatečná, přesto je zde uvedena, jelikož z ní vychází pokročilejší algoritmy. Tuhost pružiny je počítána jako:

$$k = J_P \cdot \frac{G}{L} \tag{2}$$

kde:

k	=	Torzní tuhost	[Nm·rad^{-1}]
L	=	Délka hřídele	[m]
G	=	Modul pružnosti ve smyku	[Pa]
J_p	=	Polární moment setrvačnosti	[kg·m^2]

Polární moment setrvačnosti J_p je počítán jako:

$$J_P = \frac{\pi}{32}(D^4 - d^4). \tag{3}$$

Vzorce použité v této kapitole byly převzaty z nápovědy softwaru MATLAB o komponentě *flexible shaft* [15].

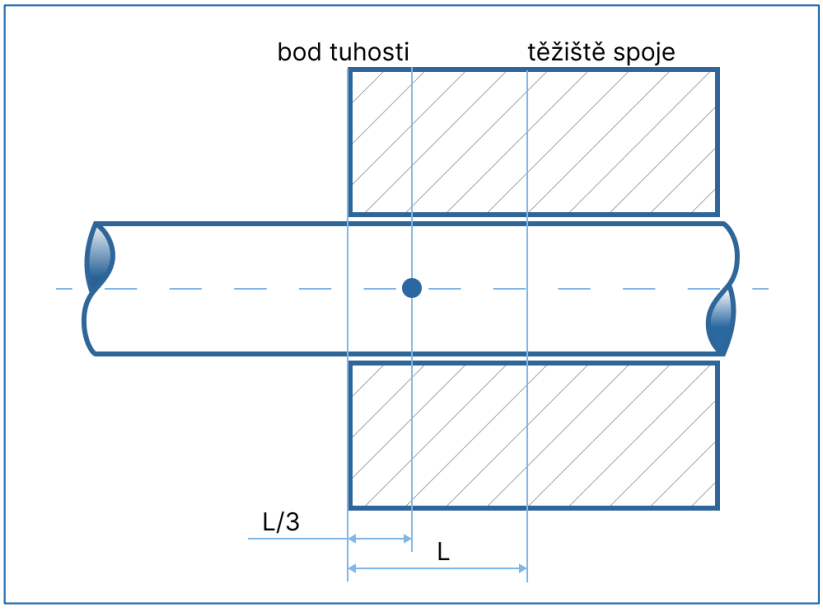
Jak již bylo zmíněno, hřídel z příkladu je modelována jako nehmotná. Nicméně některé situace vyžadují preciznější model, který využívá pokročilejší algoritmy. Pro případ, že moment setrvačnosti hřídele nelze zanedbat, je hřídel rozdělena do dvou disků, kdy každý disk má polovinu celkového momentu setrvačnosti. Tyto disky jsou spojené pružinou o tuhosti definované vzorcem (2). Je nutné uvažovat, že v takovém případě by měla mít hřídel konstantní parametry po celé délce jako jsou hustota, vnější a vnitřní průměr. V opačném případě, nebo při požadavku na přesnější model, je potřeba hřídel rozdělit na více částí. Obecně platí, že s rostoucím počtem segmentů roste i přesnost modelu [8; 15].

Jednoduchá spojka by měla být modelována také jako dva disky spojené torzní pružinou. Torzní tuhost a celkový moment setrvačnosti lze běžně získat z katalogového listu od výrobce. Literatura se dále podrobněji zabývá modelováním složitějších prvků, jako jsou nelineární spojky, ozubená kola nebo hřídele, jejichž parametry nejsou po délce konstantní (např. kuželový hřídel, hřídel s excentrickým vnitřním průměrem). Většina těchto pokročilejších modelovacích technik vychází ze zde popisovaného postupu, který je dostatečným podkladem pro model v kapitole 5. Detailnější modelovaní již nemělo vliv na vyšší přesnost modelu [8; 16].

2.4 Bod tuhosti

Následující text vychází z předchozí podkapitoly popisující modelování diskového modelu. Bod tuhosti je definován na hřídeli, na němž je namontována armatura. Model uvažuje, že za tímto bodem se hřídel stává nekonečně tuhým. Přístup umožňuje detailnější modelování tlakového spojení hřídele se spojkou. Takový spoj je podle literatury uvažován stejně jako *hřídel s nalisovanými koncovkami*. Bod se pak nachází v třetině vzdálenosti od počátku nasazení po těžiště nasazené armatury (viz obrázek 4). U jiných typů spojů je tato vzdálenost definována jinak [16].

Takto získaný bod tuhosti je pouze odhadem a slouží pro zjednodušení výpočtu spojení hřídele s armaturou. Přesnější určení bodu tuhosti by vycházelo z frekvenčních testů získaných na reálném systému [16].



Obrázek 4 Bod tuhosti podle [16]

2.5 Modelování blokovými schématy

Kapitola se zabývá praktickým vývojem modelů vycházejícího z modelovaní se soustředěnými prvky popsaného v kapitole 2.2. Modelovaný systém je tvořen souborem bloků, které představují jednotlivé funkce, objekty nebo prvky. Bloky jsou propojeny spojnicemi na základě pravidel, která reprezentují vztahy mezi jednotlivými bloky. Modelování se dělí dle způsobu na *kauzální* a *akauzální* podle vlastností bloků a toku signálu mezi nimi. Zmíněný způsob modelování využívají simulační nástroje založené na metodě soustředěných prvků.

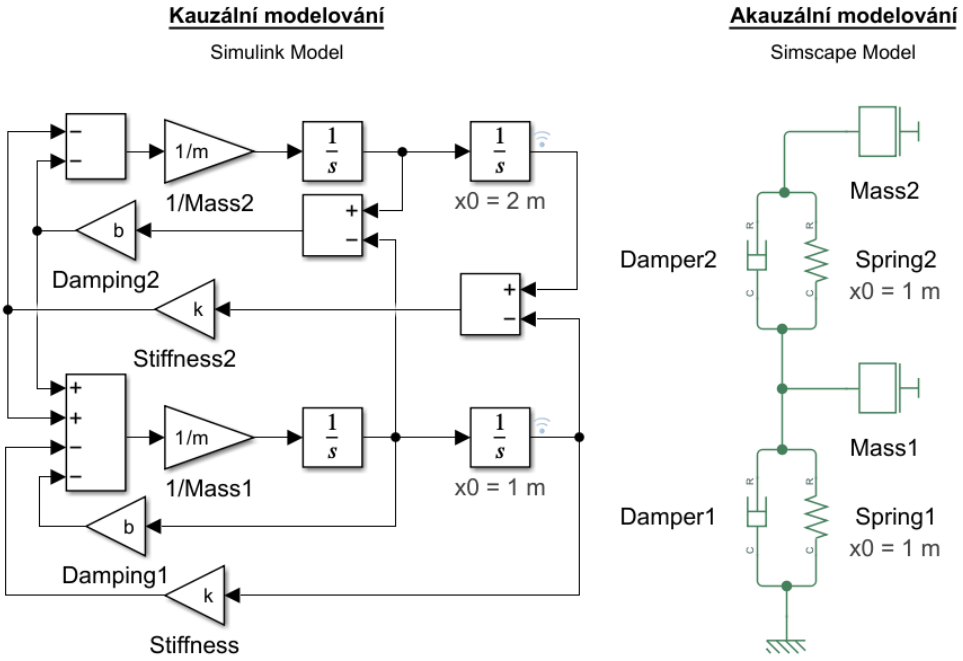
Následující kapitoly uvádějí základní popis obou přístupů k modelování a jejich vzájemné porovnání.

2.5.1 Kauzální modelování

Kauzální přístup k modelování je starší než akauzální. Vychází z modelovacích metod požívaných na analogových počítačích v minulém století. Elektronické součástky analogových počítačů představovaly matematické operace. U kauzálního modelování se k vyjádření matematických operací používají mezi sebou propojované bloky. Signál mezi propojenými bloky proudí pouze jedním směrem, což je hlavní rozdíl oproti akauzálnímu přístupu. Stejnosměrný signál reprezentuje časově proměnné číslo. Na signálu přivedeném na vstup bloku je provedena definovaná matematická operace a výsledkem operace je výstupní signál bloku [12; 17].

Obrázek 5 Kauzální přístup k modelování podle [12]

V prvních fázích vývoje modelu je potřeba matematicky popsat modelovaný systém. Po sestavení diferenciálních rovnic popisujících dynamický systém je možné sestavit blokový diagram v modelovacím prostředí. V případě modelování složitějších dynamických systémů by kauzální přístup modelování vyžadoval rozsáhlou analýzu, aby mohl být systém plně popsán [17; 12].



Obrázek 6 Srovnání kauzálního a akauzálního modelování [18]

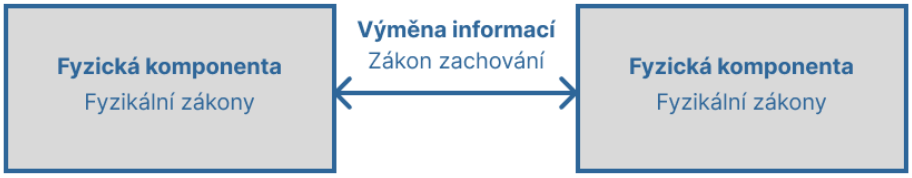
Kauzální přístup modelování není objektově orientovaný. Na obrázku 6 je uvedeno srovnání kauzálního a akauzálního modelování dvou *mass-spring-damper* systémů zapojených za sebou. Absence objektově orientovaného charakteru modelování však nemusí být chápána negativně. V určitých případech to pro daný model nemusí být nijak omezující. Kauzální přístup se často využívá pro modelování řídicích systémů a samotný Simulink na tomto přístupu funguje.

2.5.2 Akauzální modelování

U akauzálního přístupu lze nahlížet na bloky jako na objekty. U přirovnání modelovacího přístupu k analogovým počítačům by bloky reprezentovaly všechny vlastnosti dané elektronické součástky, ne pouze jednu matematickou operaci. Blokové schéma akauzálního modelování se pak více podobá reálnému systému [12].

Jak již bylo zmíněno v předchozí kapitole, u akauzálního přístupu proudí signál mezi bloky obousměrně. Jinými slovy, bloky si mezi sebou vyměňují informace o tom, jak zacházet s fyzikálními veličinami [12].

Akauzálního přístupu využívá široce používaný jazyk Modelica, který je standardizován. Na jazyku Modelica jsou založeny modelovací nástroje Dymola, MapleSim anebo zdarma dostupný nástroj OpenModelica.



Obrázek 7 Akauzální přístup k modelování podle [12]

2.6 Validace a verifikace

Součástí vývoje modelu je verifikace a validace. Oba procesy mohou využívat specifické techniky v závislosti na typu vyvíjeného modelu, avšak smysl činností zůstává neměnný. Názvy ale bývají často zaměňovány, i když mají odlišné účely a postupy [19].

Verifikací je zjišťováno, jestli samotný model pracuje, jak je požadováno. Zahrnuje například proces ladění modelu, správnou konfiguraci řešiče nebo kontrolu toku signálů. Z logického a matematického hlediska lze pak kontrolovat, jestli model nevrací nesmyslné nebo neobvyklé hodnoty proměnných [20; 19].

Validace pak zjišťuje, jak přesnou je verifikovaný model reprezentací požadovaného systému. V případě Digitálního dvojčete je model srovnáván se reálným protějškem. Vzhledem k tomu, že model je zjednodušením modelovaného systému, nelze dosáhnout dokonale přesné kopie. Posouzení přesnosti je pak individuální pro každý modelovaný případ [19; 20].

Součástí validace je získání dat z reálného světa. Podstatná jsou nejen výstupní ale i vstupní data, aby byl model simulován za stejných podmínek. Jsou situace kdy některá data nelze získat. V takovém případě se při vývoji modelu využívají různé identifikační techniky pro určení neznámých parametrů. Může nastat i opačný případ, kdy je k dispozici velké množství dat. Potom je nutné vybrat pouze ta relevantní. Touto problematikou se zabývá obor strojového učení a umělé inteligence [19].

Validace a verifikace jsou tedy důležité fáze během vývoje modelu. Neverifikovaný model může generovat nesmyslný výstup nebo v horším případě bude generovat chyby, které zůstanou nepovšimnuty. Nevalidovaný model pak může být podkladem pro chybná rozhodnutí [19].

3 MATLAB & SIMULINK

MATLAB je komerční nástroj s platformou Simulink pro fyzikální modelování. Samotný MATLAB kromě možnosti fyzikálního modelování zapadá do celého kontextu vývoje modelu. Díky řadě rozšíření lze software využít od návrhu požadavků a architektury, až po simulaci, testy a validaci.

V této kapitole budou popsány nástroje a komponenty, které jsou významné pro projekt v kapitole 5.

3.1 MathWorks a HUMUSOFT

Nástroje Matlab a Simulink jsou vyvíjené společností MathWorks. Původní MATLAB kolem roku 1976 byl jednoduchým nástrojem pro interaktivní manipulaci s maticemi, který umožnil studentům experimentovat bez nutnosti psaní a kompilování programů. Současné verze nabízejí rozsáhlé sady toolboxů pro širokou škálu disciplín, od inženýrství a matematiky po biologii a umělou inteligenci [21].

V České republice zastupuje MathWorks firma HUMUSOFT, která je oficiálním distributorem a také se podílí na vývoji nadstaveb systému MATLAB a Simulink.

3.2 Simulink a Workspace

Simulink je nástroj pro modelování, simulaci a analýzu dynamických systémů, které jsou v Simulinku tvořeny blokovými schématy. Simulink využívá kauzálního způsobu modelování. Dostupné knihovny pak obsahují převážně bloky s matematickými operacemi a prostředky pro zpracování a vizualizaci spojitého i diskrétního signálu. Dále do simulinkového modelovacího prostředí lze vkládat bloky, které jsou součástí jiných toolboxů, např. System Identification Toolbox nebo Control Systém Toolbox.

Simulink využívá pro ukládání hodnot proměnných pracovní prostory *Base workspace* a *Model workspace*. V *Base workspace* jsou uchovávány proměnné vytvořené pomocí skriptu nebo příkazového řádku. Uloženy jsou zde po celou dobu zapnutého programu MATLAB, pokud nejsou vynuceně smazány. Pracovní prostor *Model workspace* je podobný *Base workspace* s tím rozdílem, že hodnoty proměnných jsou viditelné pouze v rámci daného modelu v Simulinku. Každý model má svůj vlastní *Model workspace*. Pokud existuje proměnná se stejným názvem jak v pracovním prostoru modelu, tak i v základním pracovním prostoru, má *Model workspace* vyšší prioritu než *Base workspace* a model použije hodnotu proměnné uloženou v *Model workspace* [22; 23].

Pro model popisovaný v kapitole 5 je využíván základní pracovní prostor, protože při vývoji byl model spouštěn pomocí skriptu. V pozdějších fázích vývoje byly simulace volány pomocí vlastní MATLAB funkce (kapitola 5.9), která používá svůj vlastní

pracovní prostor. Pro odeslání požadovaných proměnných z pracovního prostoru funkce do *Base workspace* byla použita MATLAB funkce *assignin()* [22; 24].

3.3 Řešiče

Základní princip fungování řešičů u simulačních nástrojů byl popsán v práci [7]. Tato podkapitola je zaměřena na výběr řešiče na základě modelovaného systému.

V Simulinku si uživatel vybírá mezi dvěma kategoriemi řešičů – s pevným a proměnným krokem. U řešičů s pevným krokem jsou stavy modelu řešeny v pravidelných časových intervalech po celou dobu simulace. Velikost kroku může uživatel nastavit. Platí, že větší velikost kroku vede k rychlejší simulaci, ale k méně přesným výsledkům [25].

Řešiče s proměnným krokem během simulace upravují velikost kroku na základě zadané tolerance chyb. Zmenšují velikost kroku, aby dosáhly vyšší přesnosti u rychle se měnících stavů modelu. Naopak zvyšují krok simulace, když se v modelu mění stavy pomalu [25].

Kromě rychlosti měnících se stavů je u obou kategorií nutné brát v úvahu, jestli jsou stavy modelu spojité nebo diskrétní. Výběr konkrétního řešiče ještě závisí na tom, jestli jsou v modelu řešeny „tuhé“ (stiff) ODR. Na základě toho se zvolí implicitní nebo explicitní řešič. Pro modely s fyzikálními komponentami jsou voleny implicitní spojité řešiče s proměnným krokem [26; 27].

Obrázek 8 Výběr řešiče podle [26; 27]

3.4 Simscape

Jak již bylo zmíněno, samotný Simulink umožňuje pouze kauzální přístup k modelování. Knihovny nadstavby Simscape odstraňují toto omezení a poskytují uživateli řady komponent z různých fyzikálních domén, které mohou mezi sebou být propojovány akauzálně. Komponenty často reprezentují reálné prvky systémů (např. rezistor, ventil) nebo fyzikální jevy (působící síla nebo moment).

Knihovny nadstavby Simscape jsou rozděleny podle fyzikálních domén. Pomocí konvertorů, které jsou v knihovnách obsaženy, je možné fyzikální domény propojovat na základě fyzikálních zákonů (například blok *Rotational Electromechanical Converter* převádí elektrickou energii na mechanický rotační pohyb). Díky možnosti propojování různých fyzikálních domén je možné s rozšířením Simscape tvořit tzv. Multidomain modely [28; 7].

Simscape je pak možné rozšířit o další knihovny. Jednou z nejvýznamnějších je *Multibody* knihovna, která umožňuje modelovat 3D systémy. V knihovně jsou obsaženy elementární prvky (válec, kvádr apod.), uživatel ale může importovat i vlastní CAD modely. Tělesům jsou přiřazeny materiálové vlastnosti a z jejich geometrie je vypočten moment setrvačnosti. Další komponenty knihovny *Multibody* určují vztahy mezi jednotlivými tělesy. Například prvky podkategorie *Joints* určují, jak se vůči sobě budou tělesa pohybovat. Pro výslednou simulaci je pak automaticky generována 3D vizualizace dynamiky systému [7; 29].

Významným rozšířením Simscape pro tuto práci je knihovna *Simscape Driveline*. V ní jsou obsaženy komponenty pro modelování translačních a rotačních mechanických systémů. Některé komponenty jsou již hotové modely prvků mechanických systémů – např. brzdy, motory, převodovky nebo větrná turbína. Při modelování průmyslových strojů může tato knihovna usnadnit vývoj komplexních částí, které by pro modelování ze základních komponent vyžadovaly hlubší analýzu [30].

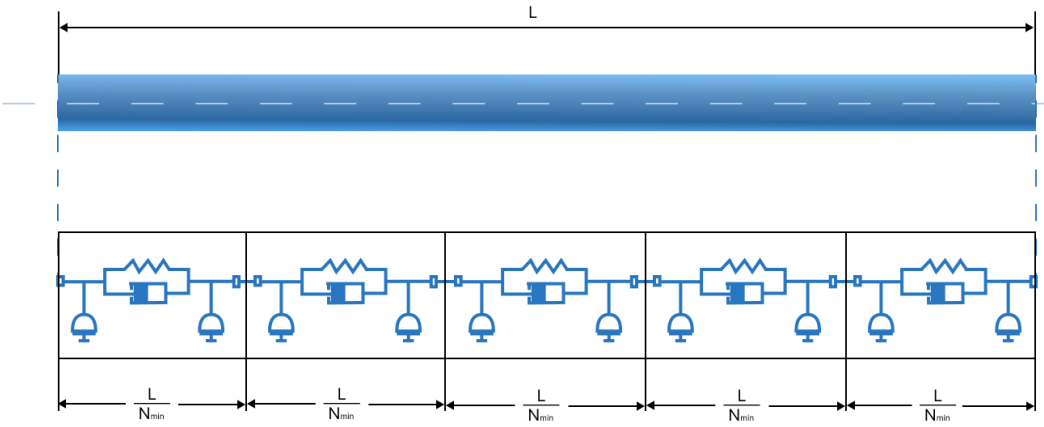
3.5 Flexible shaft

Tato podkapitola bude vycházet ze zdroje [15], kde je komponenta *Flexbile Shaft* detailně popsána. Zde je vysvětlen základní princip bloku a jeho návaznost na kapitolu 2.3 o modelování diskové soustavy. Komponenta *Flexible Shaft* je stěžejním prvkem modelu projektu popisovaném v kapitole 5.

Flexible Shaft je součástí Simscape knihovny Driveline – Couplings and Drives. Jak naznačuje název, komponenta umožňuje modelovat flexibilní hřídel. A to nejen torzně, ale i v ohybu. Pozornost zde bude věnována převážně krutu vzhledem k tomu, že model v projektu se zabývá analýzou torzních vibrací.

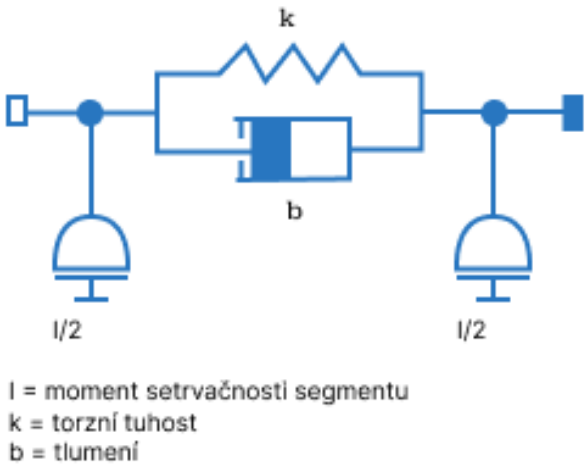
Obrázek 9 Blok Flexible Shaft v knihovně Simscape Driveline

Torzní model je sestaven z pružinových tlumičů sériově zapojených se setrvačnostmi. Setrvačnosti zde reprezentují disky propojené pružinami, jak bylo popsáno v kapitole 2.3. Minimální počet takto zapojených pružinových tlumičů je definován parametrem „Minimum number of flexible elements“ $\mathbf{N_{min.}}$



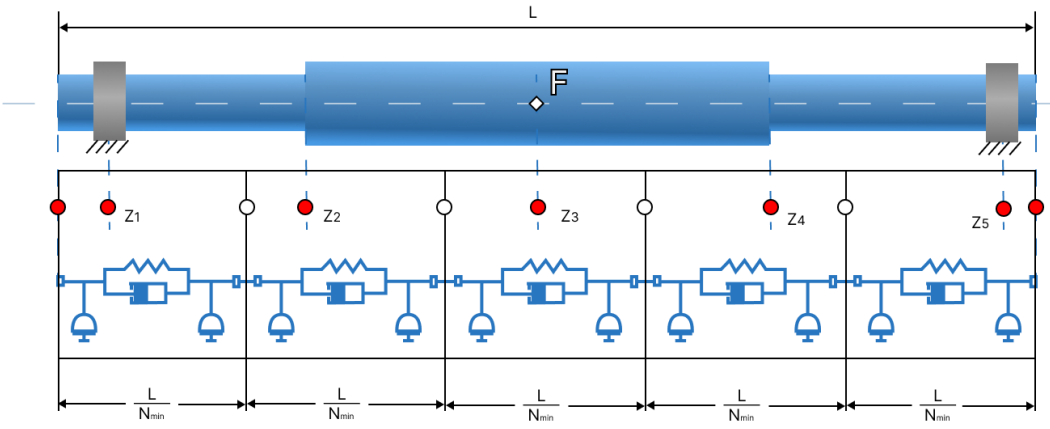
Obrázek 10 homogenní hřídel a jemu ekvivalentní torzní model

Jednoduchý systém pružinového tlumiče se setrvačností je modelován metodou soustředěných parametrů (viz kapitola 2.2). Vychází z dříve popisovaného diskového modelu, zde je ale uvažováno tlumení na rozdíl od dříve popisované základní analýzy. Systém je definován parametrem tlumení, parametrem torzní tuhosti (pružina) a setrvačností, která je rovnoměrně rozdělena do dvou částí.



Obrázek 11 Systém torzního pružinového tlumiče

Na obrázku 10 je znázorněn model hřídele, jehož parametry se nemění po celé jeho délce. Komponenta *Flexible Shaft* ale umožňuje modelovat komplikovanější případy, kdy se rozměry hřídele mění nebo na něj působí externí zátěž. Hřídel se v takovém případě rozdělí do segmentů podle L/N_{min}, stejně jako na obrázku 10. V oblastech hřídele, kde dochází ke změně rozměrů, působení externích sil nebo kde je hřídel uložena v ložisku, je umístěn abstraktní uzel. Místa abstraktních uzlů jsou označena jako Z_n.



Obrázek 12 Umístění abstraktních uzlů

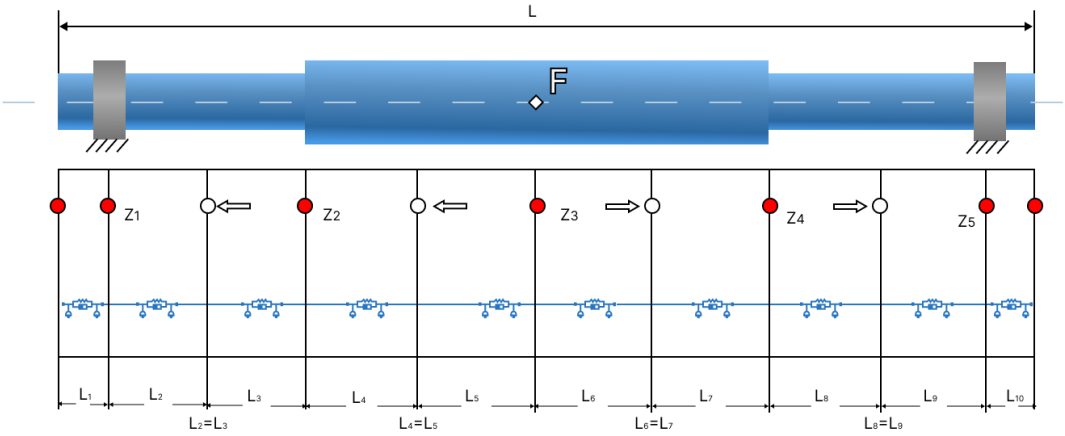
U hřídele je potřeba zvolit segmentaci hřídele podle tuhostí a setrvačností, nebo podle materiálu a geometrie. U segmentace podle materiálu a geometrie je hřídel parametrizována délkami a průměry segmentů. Materiál je určen hustotou a modulem pružnosti ve smyku, přičemž parametry jsou homogenní po celé hřídeli. Komponenta *Flexible shaft* si z těchto parametrů dopočítá setrvačnosti a torzní tuhosti podle vzorců v kapitole 2.3. V druhém případě, u parametrizace podle tuhostí a setrvačností, jsou segmenty definovány přímo těmito parametry. To má vhodné využití například tehdy, když hřídel nemá homogenní materiál po celé délce.

Umístění ložisek je možné definovat pouze u parametrizace podle materiálu a geometrie. Externí síla má vliv pouze v případě, že je u komponenty uvažován ohyb.

Se stanovenou segmentací hřídele jsou upraveny pozice abstraktních uzlů Z_n, na jejichž základě je určen počet podsystémů pružinových tlumičů (viz obrázek 13). Uzly znázorněné bílou barvou jsou přesunuty na pozice:

$$\mathrm{L_n} = \frac{(z_n - z_{n+1})}{2} \tag{4}$$

kde umístění L_n jsou znázorněna na obrázku 13.



Obrázek 13 Model hřídele s konečným počtem pružinových tlumičů

Parametry segmentů jsou zadávány v polích. Záleží na pořadí prvků v poli. Při vztažení této skutečnosti k obrázku 12, přísluší první prvek pole segmentu nejvíce vlevo (okraj hřídele až Z_1). Jak již bylo zmiňováno v kapitole 2, s rostoucím počtem podsystémů pružinových tlumičů poroste i přesnost modelu. S tím ale poroste i náročnost simulace.

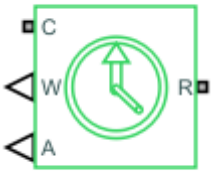
Pro analýzu torzních vibrací lze komponentou *Flexible Shaft* reprezentovat jakoukoliv rotační součást, pokud ji lze definovat výše popsanými způsoby.

3.6 Ideal Rotational Motion Sensor

Snímač *Ideal Rotational Motion Sensor* je součástí základní knihovny Simscape. Blok snímá veličinu mezi dvěma mechanickými rotačními uzly a převádí ji na fyzikální signál, kterým může být informace o úhlu, úhlové rychlosti nebo zrychlení. Vlastností „ideal“ je myšleno to, že nezohledňuje tření, setrvačnost, zpoždění apod [31].

Viditelnost portů pro snímaní polohy, rychlosti a zrychlení se nastavuje v parametrech bloku. Také lze nastavit mód měření – absolutní nebo relativní. U relativního měření je měřen rozdíl rychlosti, zrychlení a polohy portu R vzhledem k portu C. U absolutního měření je port C neviditelný a měřené veličiny jsou vztaženy k referenčnímu uzlu (k zemi – blok *Mechanical Rotational Reference*) [31].

V projektu (kapitola 5) byl tento blok vložen pro snímaní vibrací a k verifikaci bridge.

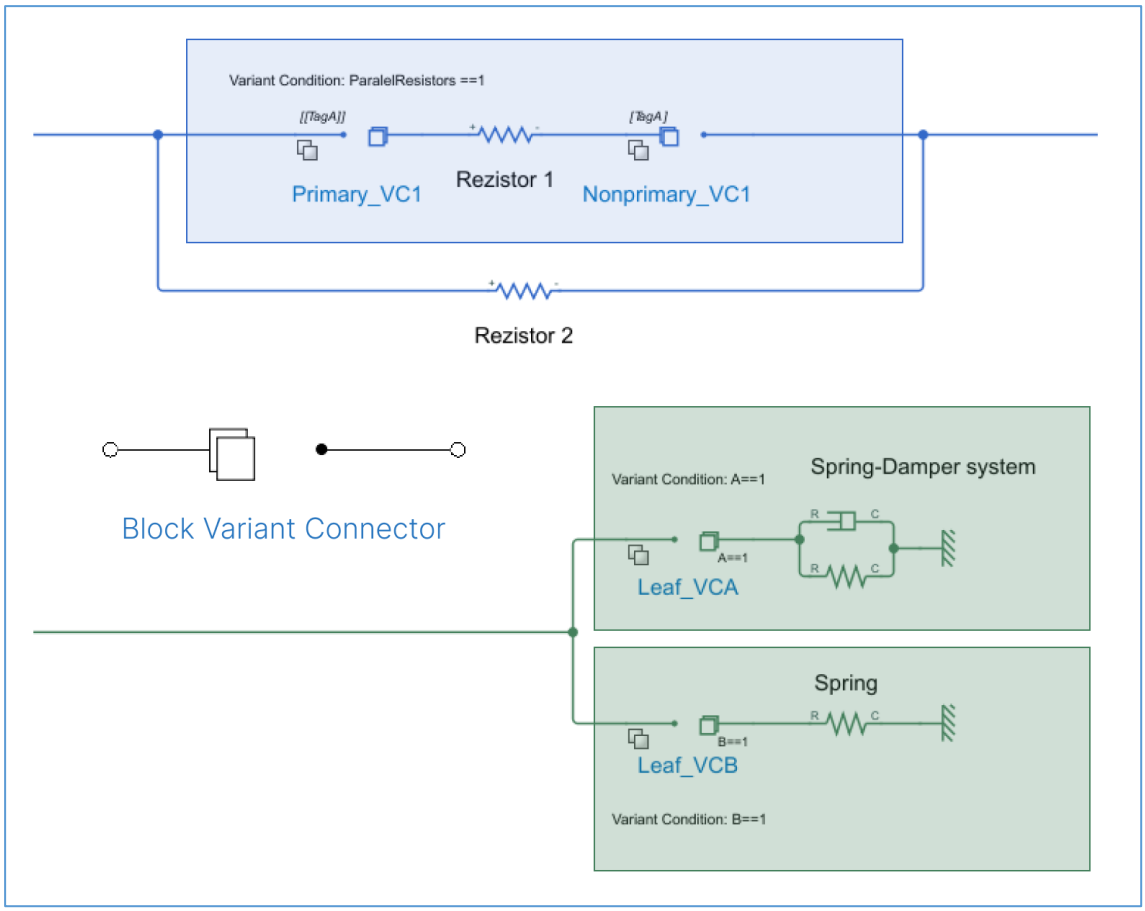


Obrázek 14 Ikona bloku Ideal Rotational Motion Sensor

3.7 Variant Connector

Vlastnosti popisovaného bloku v této podkapitole vychází z MATLAB nápovědy [32]. Bloky *Variant Connector* umožňují vytvořit více variant blokového schématu v jednom modelu. To platí pouze pro fyzikální síť modelu, tedy síť sestavenou z bloků knihovny Simscape. U sítě využívající Simulink signál lze použít *Variant Subsystem*. Funkce je podobná, ale pravidla pro zapojení v blokovém schématu jsou odlišná vzhledem ke kauzálnímu přístupu u modelování se Simulink signálem.

Na základě splnění definovaných podmínek bloku je možné odpojit určitou oblast nebo větev bloků od zbytku blokového schématu. Efekt je stejný, jako by odpojené bloky byly odstraněny bez nutnosti manuálního odebrání před kompilací modelu, ale na základě splnění podmínek.



Obrázek 15 Zapojení bloků Variant Connector

Na obrázku 15 jsou uvedeny dva způsoby zapojení, které je nutné v bloku nastavit. Zelené oblasti jsou tzv. listy. Síť za bokem *Variant Connetor* se při nesplnění podmínky celá „vypne“. Při tomto zapojení nesmí být modelována smyčka, která by síť za blokem připojila zpět před blok. V uvedeném příkladu bude odpojena pružina (list B) nebo systém pružina-tlumič (list A), pokud se proměnné *A* nebo *B* nebudou rovnat hodnotě TRUE.

U elektrického obvodu, značeného modrou barvou, se ze zapojení odpojí paralelně zapojený rezistor, pokud se proměnná *ParallelResistors* nebude rovnat hodnotě TRUE. Zapojení odpojí určitou oblast obvodu ohraničenou *Primary* a *Nonprimary* nastavenými bloky *Variant Connector*. U *Primary* bloku se nastavuje podmínka pro připojení oblasti do sítě. Blok *Variant Connector* mění barvu podle typu fyzikální sítě, do které je připojený. Zelená síť v obrázku je mechanická, modrá síť je elektrická. Při zapojování bloku je nutné brát v úvahu jeho orientaci.

V projektu jsou bloky *Variant Connector* použity pro různé typy motorů, konstrukce osy a pro možnost nasazení *bridge* na barevník. Bloky jsou v projektu aktivovány na základě BOOL proměnných.

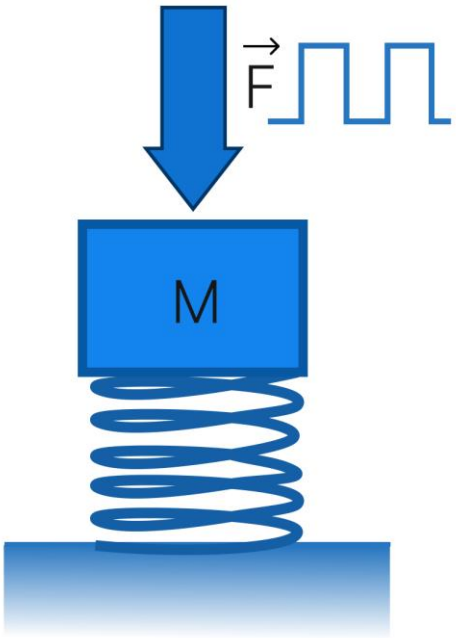
4 TVORBA MODELU S VYUŽITÍM SIMSCAPE

V této kapitole budou demonstrovány základní postupy při tvorbě modelu s využitím knihoven Simscape. Jednoduchý model a hodnoty jeho parametrů jsou převzaty od společnosti MathWorks [33].

4.1 Popis modelovaného systému

Popisovaný příklad demonstruje propojení jednoduchého akauzálně modelovaného systému *Mass-Spring-Damper* s kauzálním řízením pomocí PID regulátoru. V realitě by se jednalo o závaží připevněné na tlačné pružině. Tento systém by byl periodicky stlačován silou regulovanou pomocí regulátoru.

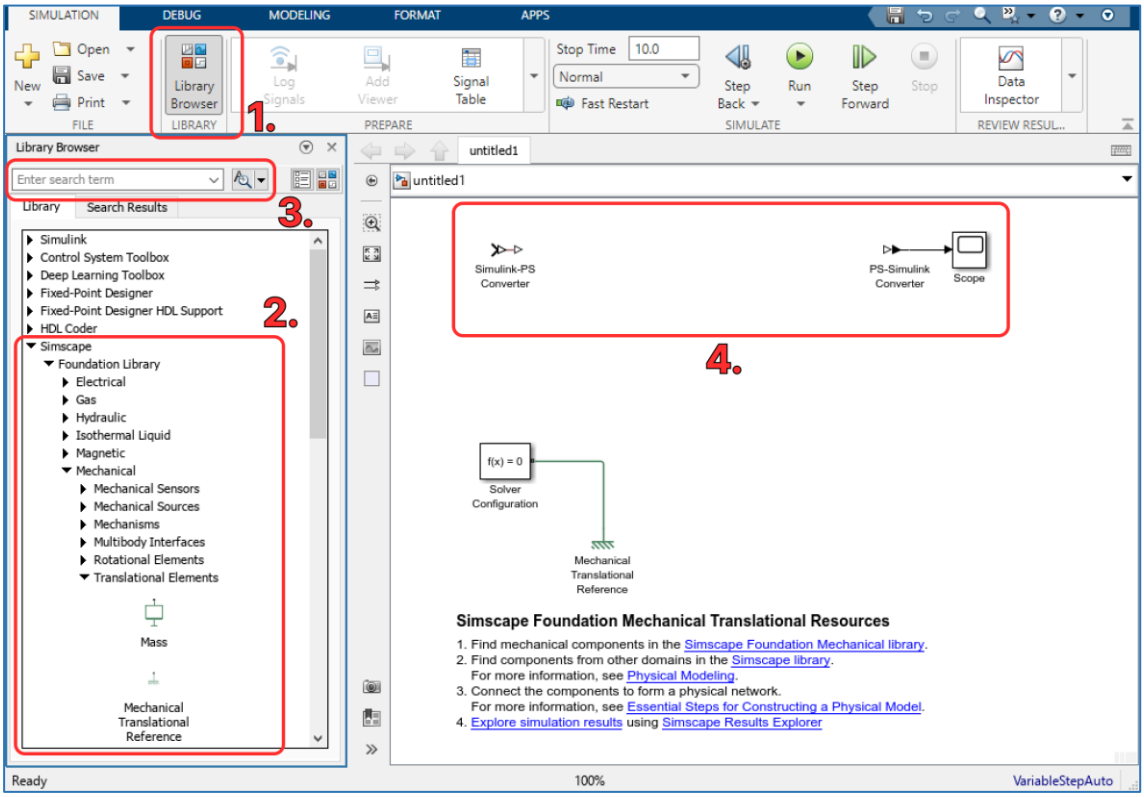
PID regulátor je jedním z nejzákladnějších prvků v řídicích systémech. Využití modelu hmoty s pružinou a tlumičem, známého jako *Mass-Spring-Damper*, je základním prvkem pro modelovaní mechanických systémů pomocí metody se soustředěnými parametry. Celý systém by mohl být modelován kauzálně bez použití knihoven nadstavby Simscape.



Obrázek 16 Modelovaný systém Mass-Spring-Damper

4.2 Blokové schéma

Modelovací prostředí lze spustit příkazem „Simulink“ v konzoli MATLABu. Otevřené okno nabízí mimo vytvoření „prázdného“ modelu také založení rozsáhlejšího projektu s verzováním, nebo modely Simscape s předdefinovanými parametry simulace podle modelované domény. Například při volbě *Mechanical Translational* se otevře model s již nadefinovaným blokem řešiče připojeného k mechanické referenci (rám, zem). Obdobným způsobem jsou zakládány Simspace modely i u jiných fyzikálních domén. Dále se v modelovacím prostředí nachází bloky, které jsou v dané fyzikální doméně často používány.



Obrázek 17 prostředí nově vytvořeného Simscape modelu

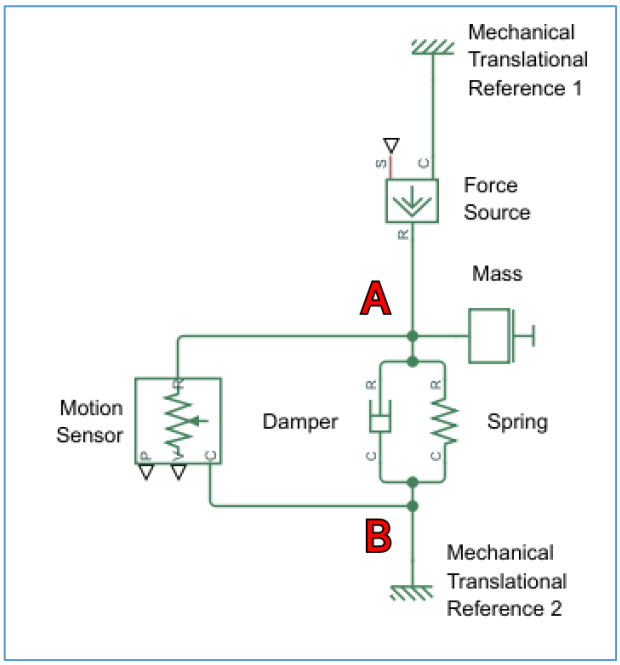
Popis prostředí na obrázku 17:

1. Otevření knihoven s komponenty (Library Browser)
2. Knihovna se Simscape komponenty (vybrána podkategorie Translational Elements)
3. Vyhledávání komponent podle názvu
4. Komponenty přidané automaticky při otevření nového modelu (v závislosti na Simscape doméně)

Je vhodné dodat, že určité knihovny jsou viditelné pouze v případě, že jsou obsaženy v aktivované licenci. Komponenty se dají vyhledávat i dvojklikem do modelovacího prostředí, které funguje stylem „drag and drop“.

Z otevřené kategorie *Translational Elements* jsou do modelovacího prostředí vloženy komponenty *Mass*, *Spring* a *Damper*, které jsou základními prvky modelovaného Simscape systému. Z kategorie *Mechanical Sources* je do modelu přidán blok *Ideal Force Source*, kterému nelze nastavit žádné parametry, pouze převádí „obecný“ fyzikální signál na sílu v translačním mechanickém Simscape schématu. Z další kategorie *Mechanical Sensors* je přidán blok *Ideal Translational Motion Sensor*, který bude snímat změnu polohy mezi dvěma translačními uzly (na obrázku 18 jsou tyto uzly označeny jako A, B). Blok má podobné vlastnosti jako *Ideal Rotational Motion Sensor* z kapitoly 3.6 [34; 35].

Do schématu je potřeba přidat ještě jeden blok *Mechanical Translational Reference*. Lze ho opět najít v knihově, nebo uživatel může zkopírovat již umístěný blok v modelovacím prostředí. Bloky jsou následně propojeny podle obrázku 18.

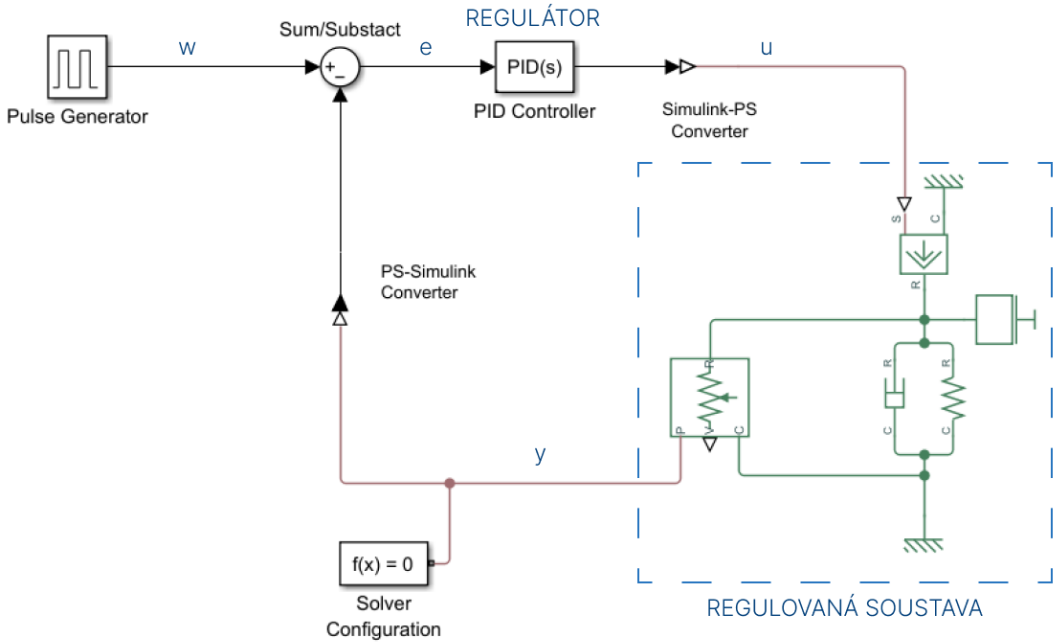


Obrázek 18 Blokové zapojení Simscape modelu

Tímto způsobem je modelovaná regulovaná soustava zvoleného regulačního obvodu. Nyní je potřeba model rozšířit o řízení a zpětnou vazbu. Do modelovacího prostředí jsou přidány bloky *Sum*, *PID Controller* a *Pulse Generator*, které se nacházejí v knihovnách kategorie Simulink. Nezbytné jsou také bloky *Simulink-PS Converter* (převádí Simulink signál na fyzikální) a *PS-Simulink Converter* (převádí fyzikální signál na Simulink signál), které se v modelovacím prostředí nacházeli při otevření modelu (oblast 4 na obrázku 17).

U Bloku *Sum* jsou dva vstupní porty, z nichž jeden musí být nastaven na zápornou hodnotu, aby regulovaná veličina byla odčítána od veličiny požadované. Alternativně lze použít blok *Subtract*, který má tímto způsobem nastavené vstupy ve výchozím stavu.

Nyní jsou v modelovacím prostředí přichystány všechny bloky pro vytvoření regulačního obvodu, který je zapojen následovně:



Obrázek 19 Zapojení bloků do regulačního obvodu

kde:

w	=	Žádaná veličina
e	=	Regulační odchylka
u	=	Akční veličina
y	=	Regulovaná veličina (pozice)

Aby mohl být model simulován, musí být ke každé fyzikální Simscape síti připojen blok *Solver Configuration*, který určuje parametry řešiče před zahájením simulace. K soustavě může být připojen, jak je uvedeno na obrázku 19.

Pro zobrazení výsledků simulace je do schématu přidán blok *Scope*, do kterého bude přiveden signál větví „w“ a „y“.

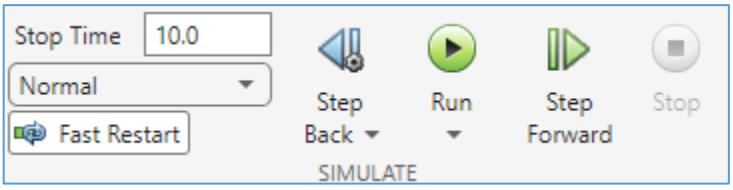
4.3 Spuštění simulace

V následujících tabulkách jsou uvedeny hodnoty parametrů, se kterými bude simulace spouštěna:

Tabulka 1 Parametry modelu

Regulovaná soustava		
Název bloku	Název parametru	Hodnota
Translational Damper	Damping coefficient	100 N/(m/s)
Translational Spring	Spring rate	400 N/m
Mass	Mass	3.6 kg
PID Regulátor		
Název parametru		Hodnota
Proporcionální složka (P)		700
Integrační složka (I)		4000
Derivační složka (D)		15
Pulse Generator		
Název parametru		Hodnota
Amplituda		0.5
Perioda		5 s
Délka pulzu		50% periody
Fázové zpoždění		3 s

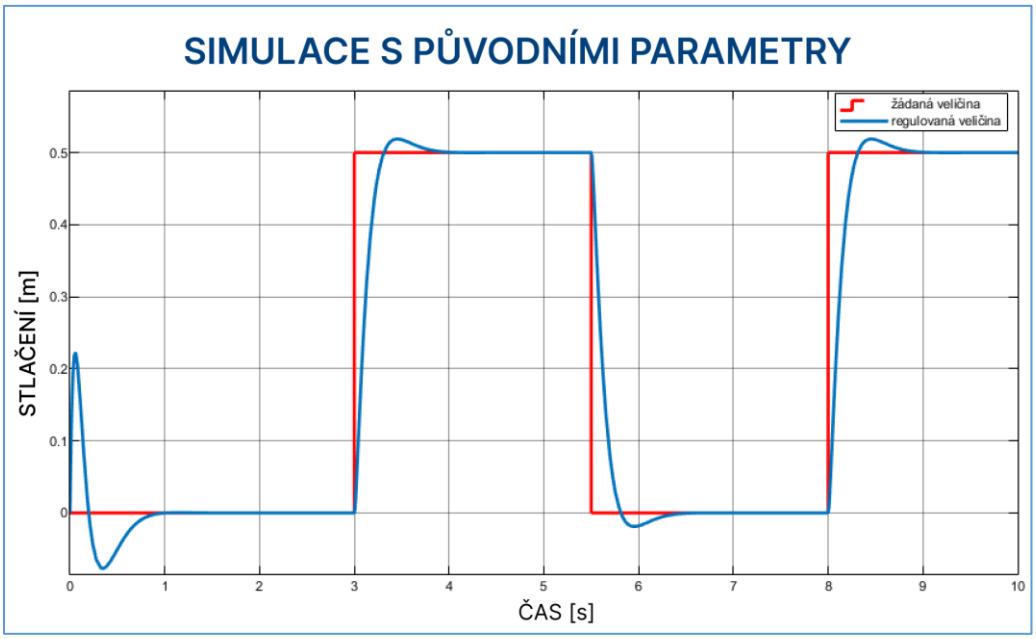
Simulace může být spuštěna přímo v modelovacím prostředí v záložce „Simulation“. V poli *Stop Time* se nastaví délka simulovaného běhu modelu, která je v tomto případě je nastavena na 10 sekund. Níže se nachází nabídka s nastavením režimu simulace, který je ve výchozím stavu nastaven na *Normal*. Režimy *Accelerator* a *Rapid Accelerator* mají význam pro komplexnější a časově náročnější simulace. Zelené tlačítko *Run* spustí kompilaci modelu a následnou simulaci. Simulace může být přerušena tlačítkem *Stop*. Tlačítka *Step Forward* a *Step Back* se používají při ladění simulace.



Obrázek 20 Oblast pro spuštění simulace

Model může být spuštěn přímo v konzoli MATLABu příkazem *sim('nazev_modelu')*. V takovém případě musí být model načten ve vyhledávací cestě (Path) MATLABu.

Na následujícím grafu jsou porovnány křivky řídícího signálu bloku *Pulse Generator* (požadovaná veličina) se stlačením pružiny (pozice – regulovaná veličina). Simulace proběhla s hodnotami parametrů v tabulce 1.



Obrázek 21 Výsledky simulace

5 VLASTNÍ MODEL

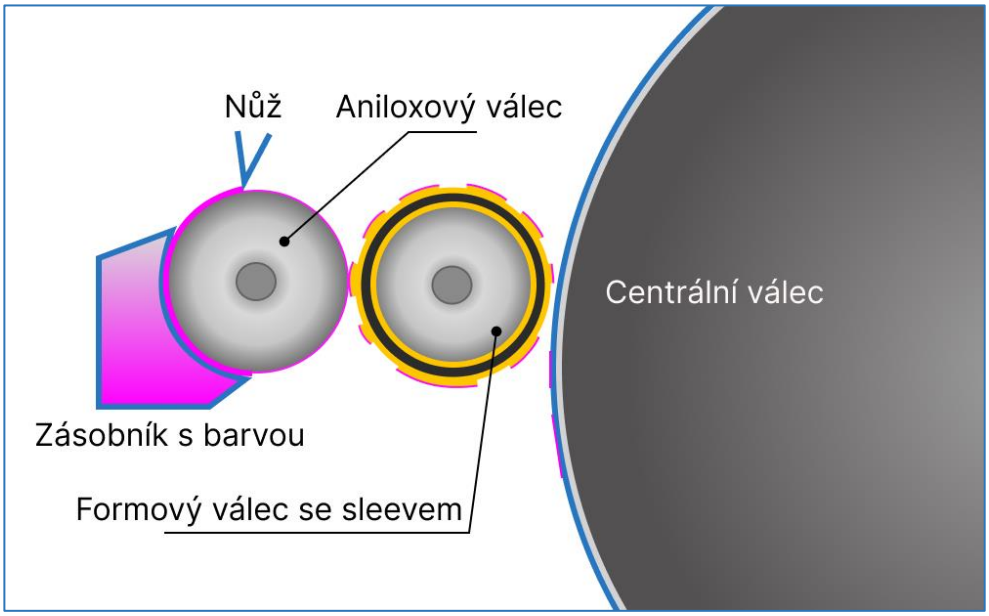
V kapitole bude detailně popsán vyvíjený model pro firmu SOMA. Podkapitoly se detailně zabývají získáním a úpravami parametrů pro stroj. Text zároveň slouží jako podklad pro případné rozšíření nebo úpravy do budoucna.

5.1 Flexotisk a motivace projektu

Flexotisk je rotační metoda tisku, často používaná pro potisk obalů na materiály jako lepenka, papír nebo fólie. Využívá plastickou tiskovou desku (formu s obrázkem), která je připevněna na válec (sleeve) nasazený na jádro formového válce. Aniloxový válec s upraveným povrchem poté přenáší barvu ze zásobníku na vyvýšené plochy tiskové desky. Střídání tiskové a netiskové plochy v místě kontaktů tiskových válců vyvolává vibrace, které mají vliv na výslednou kvalitu tisku.

Jedním z nežádoucích jevů vibrací je torzní kmitání tiskového válce, které má přímý dopad na kvalitu tisku a regulaci polohy válce. Správným navržením osy tohoto válce lze minimalizovat negativní účinky vibrací a dosáhnout přesnější regulace jeho polohy.

Cílem vývojového projektu je vytvořit novou metodiku pro návrh tiskových válců, která umožní sledovat chování osy tiskového válce již během návrhu, predikovat výslednou kvalitu tisku a provádět virtuální testování různých variant rotační osy. Proto bylo vyvinuto digitální dvojče osy formového válce, které umožňuje analýzu vlastních frekvencí již ve fázi návrhu.

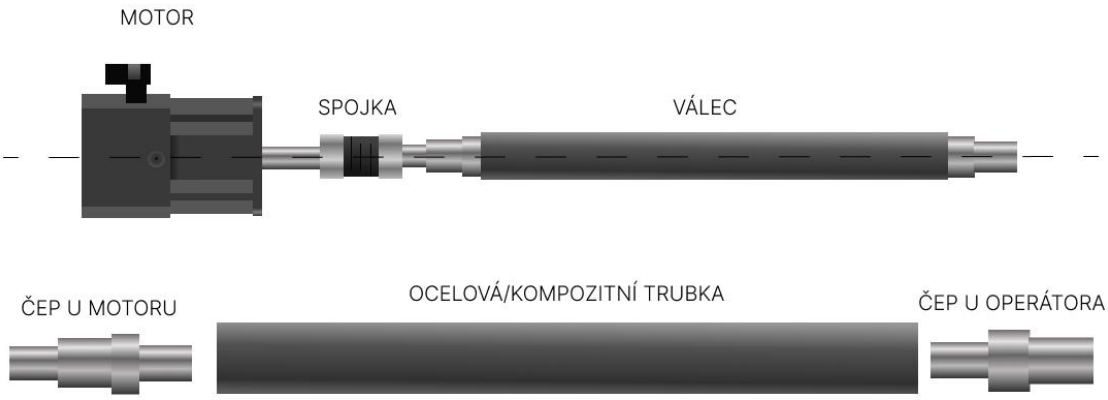


Obrázek 22 Schéma procesu flexotisku

5.2 Popis modelovaného systému

Modelovaným systémem je část barevníku fexotiskového stroje. Základními komponentami toho systému jsou pohon, spojka a formový válec, který je sestaven ze dvou čepů a trubky. Schéma je znázorněno na obrázku 23.

Jedním z cílů projektu je, aby byl model do určité míry univerzální. To z toho důvodu, že firma SOMA vyrábí a testuje různé kombinace typů zmiňovaných komponent v závislosti na typu stroje. Pro motor jsou uvažovány primárně produkty od společnosti Bosch Rexroth, konkrétně řady MS2N10. Jedná se o synchronní servomotory. Koncové číslo „10“ označuje katalogovou velikost motoru. Pro model jsou podstatné mechanické parametry motoru, tedy moment setrvačnosti rotoru a rozměry hřídele motoru. Moment setrvačnosti je rozdílný pro různé typy motorů z této řady, rozměry hřídele jsou u všech produktů z řady MS2N10 stejné [36].



Obrázek 23 Schéma modelovaného systému

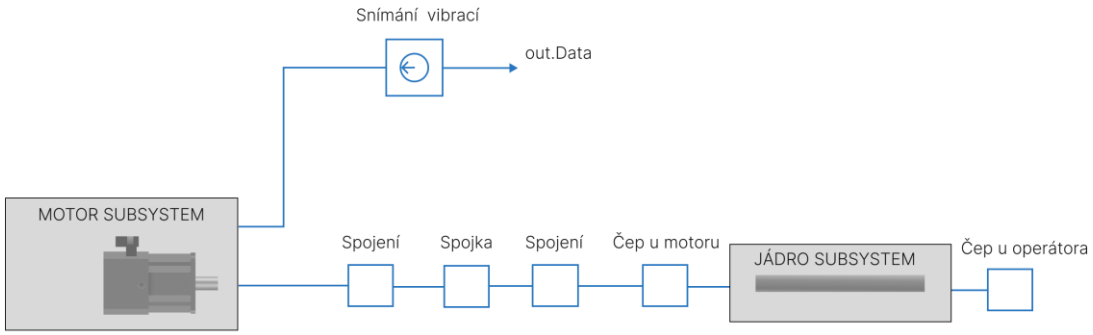
Pro model budou uvažovány vlnovcové spojky, které firma běžně používá. Materiálové parametry a rozměry jsou získány z katalogu výrobce. Problematika modelování spojky pro tento systém je podrobněji popsána v kapitole 5.8. Při vývoji modelu byla snaha využívat co nejvíce katalogová data od dodavatelů. To by umožnilo rychlé testování různých kombinací komponent bez nutnosti větší komunikace s dodavateli.

Čepy si firma vyrábí sama. V práci jsou vynechány detaily o přesné konstrukci čepů. Pro účely popisu přibližného popisu je navržen vlastní design, který se na reálném stroji nepoužívá. Stejně tak bude uvažována trubka formového válce, která je buď vyráběna firmou nebo dodavatelem. Materiálové vlastnosti pro trubku jsou získány z výkresů firmy nebo od dodavatele.

5.3 Simscape model

Na následujícím obrázku je zjednodušeně vyobrazeno blokové schéma modelovaného systému v Simscape prostředí. Snímek z modelovacího prostředí je v příloze I. Obrázek 24 je pouze ilustrativní pro přehled, jak jsou mezi sebou propojeny komponenty, které jsou popsány v pozdějších kapitolách.

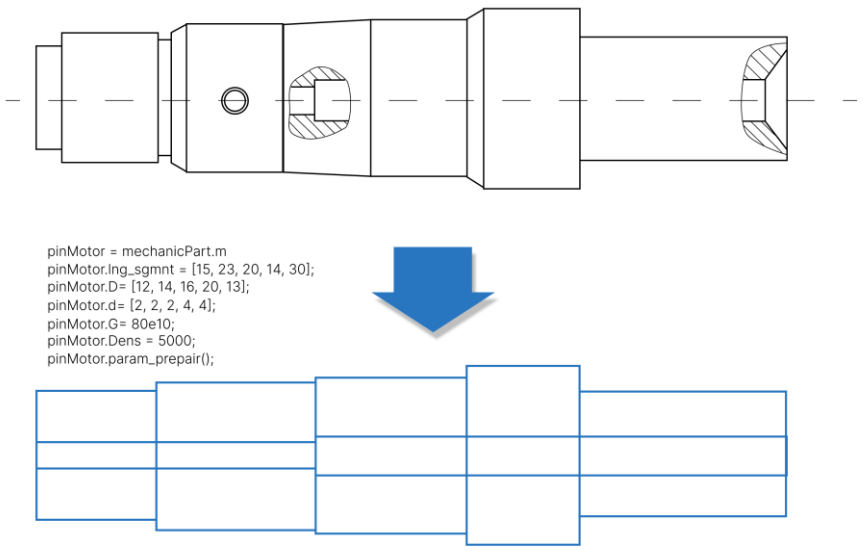
Subsystémy motoru a jádra obsahují různé varianty pro zapojení bloků podle požadavků na simulaci (například simulace s bridgem).



Obrázek 24 Zjednodušené schéma bloků v Simscape

5.4 Modelování čepů a jádra

Detailnější konstrukce čepu zde nebude popisována. Pro účely práce si lze čep představit jako odstupňovaný hřídel s vnitřním vrtáním tvořený stejným materiálem po celé jeho délce. V modelu jsou čepy reprezentovány Simscape bloky *Flexible shaft*. Čepy je nutné rozsegmentovat a vhodně aproximovat (viz kapitola 3.5). Radiální vrtání a malá zkosení jsou zanedbána.



Obrázek 25 Aproximace čepu u motoru, hodnoty jsou fiktivní

Pro přehlednější prezentaci parametrů čepu v modelu byla vytvořena třída *mechanicPart*. Proměnné třídy jsou parametry potřebné pro definování čepu blokem *Flexible shaft*. Přehled proměnných je uveden v tabulce 2. Parametry *D*, *d*, *lng_sgmnt*, *Dens* a *G* jsou nutné pro definování čepu, ostatní parametry jsou dopočítány metodou bloku *param_prepair()*. Metoda *countSegment()* spočítá segmenty a uloží je do proměnné *n_segment*. Při zadávání rozměrů do polí je nutné respektovat pořadí segmentů vzhledem k blokovému zapojení komponent *Flexible shaft* v Simulinku. Čepy jsou pojmenovány podle jejich pozic v barevníku. Čep u motoru se nazývá *pinMotor* a čep u operátora *pinOperator*. Třída *mechanicPart* je univerzální a je využita i při modelování dalších částí, které jsou definovány stejnými parametry jako čepy.

Tabulka 2 Parametry třídy mechanicPart

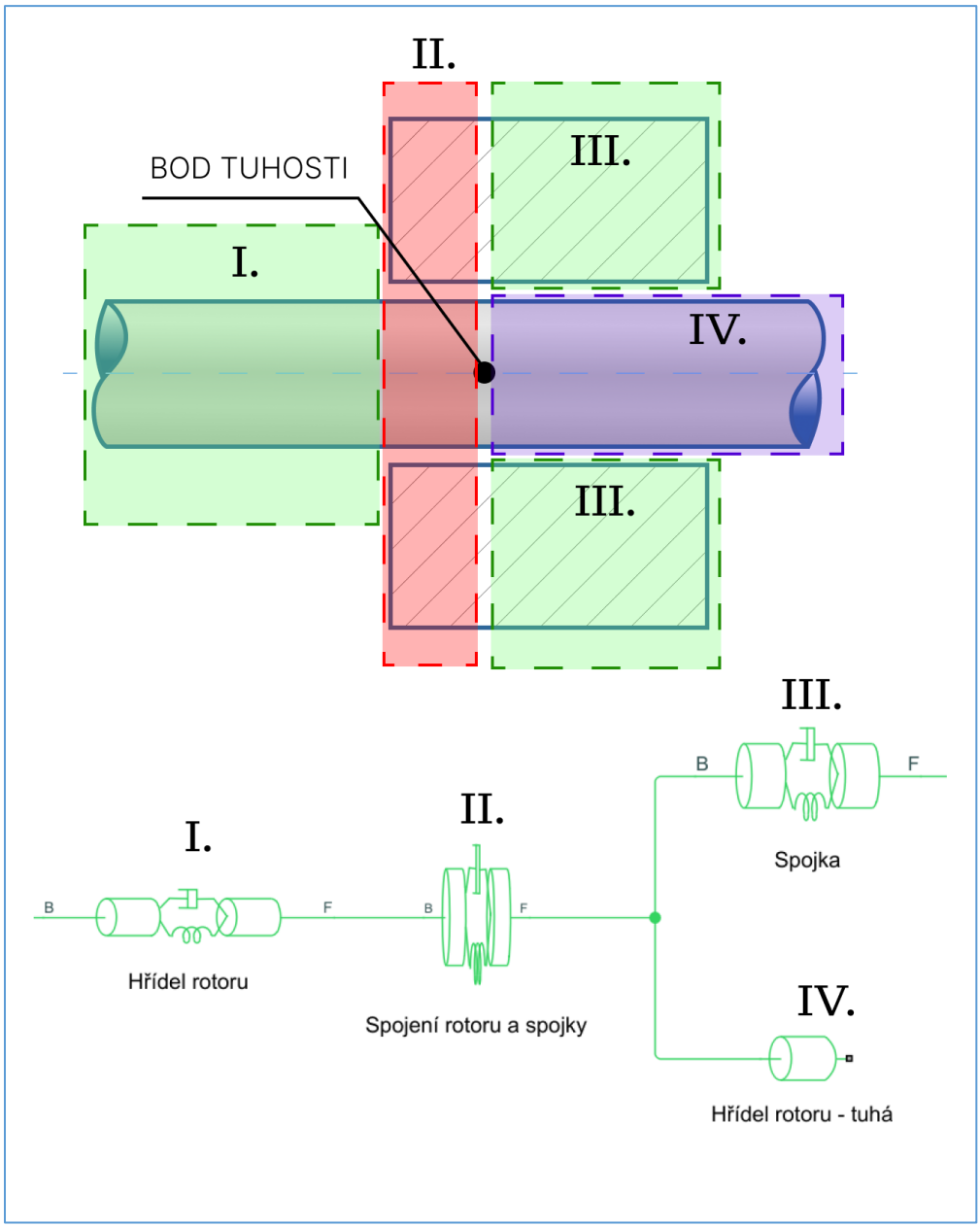
Název vstupu	Popis
D	Pole vnějších průměrů segmentů [mm]
d	Pole vnitřních průměrů segmentů [mm]
lng_sgmnt	Pole délek segmentů [mm]
Dens	Hustota materiálu [kg·m^{-3}]
G	Modelu pružnosti ve střihu [Pa]
n_segment	Počet segmentů
m	Hmotnost [kg]
tor_stf	Torzní tuhost [Nm/rad]
J	Moment setrvačnosti [kg·m^{2}]

Na jádro je možné nahlížet z pohledu modelování stejně jako na jednodušší čep. Jeho vnější průměr je konstantní po celé délce, může se měnit jeho vnitřní průměr. Při převodu do třídy *mechanicPart* bylo využito maximálně tří segmentů, a to u případu, kdy se měnil vnitřní průměr v místech spojení jádra s čepem. V Simulinku je jádro modelováno v subsystému *Composite* (v rané fázi vývoje bylo uvažováno, že je jádro vyrobeno pouze z kompozitního materiálu), kde se nachází dvě varianty zapojení – samotná trubka reprezentovaná jedním blokem *Flexible shaft* a trubka s nasazeným bridgem (kapitola 5.11). Výběr varianty modelu je řešen pomocí zapojení bloků *Variant Connector*.

Obrázek 26 Varianty jader v subsystému Composite (aktivní jádro s bridgem)

5.5 Modelování spojení hřídel – armatura

Struktura modelu obsahuje čtyři mechanická spojení, které lze typově přirovnat ke spojení hřídele a armatury. Pojmenování i způsob, jakým je spojení modelováno, vychází z kapitoly 2.4. Model uvažuje, že část hřídele se od určitého bodu stává nekonečně tuhou. Tato část hřídele je modelována blokem *Inertia*, který reprezentuje moment setrvačnosti. Část spojení od hrany armatury nasazené na hřídeli až po *bod tuhosti* je modelována jako součet vlastností hřídele a armatury. Obrázek 27 znázorňuje toto zapojení schematicky v porovnáním se zapojením pomocí Simscape bloků. Tuhý konec hřídele je paralelně zapojen se spojkou, která v tomto případě reprezentuje armaturu.



Obrázek 27 Schéma modelovaného spojení

Pro výpočet parametrů spojení byla napsána funkce *connection_point()* s parametry:

Tabulka 3 Vstupní parametry funkce connection_point()

Název vstupu	Popis
Shaft	Třída hřídele
Idx	Index segmentu hřídele, na kterém je realizován spoj
Hub	Třída armatury
hubIdx	Index segmentu armatury, na kterém je realizován spoj
contact_area	Délka styku [mm]
Ratio	Poměr vzdálenosti od začátku armatury po její těžiště

Tabulka 4 Výstupní parametry funkce connection_point()

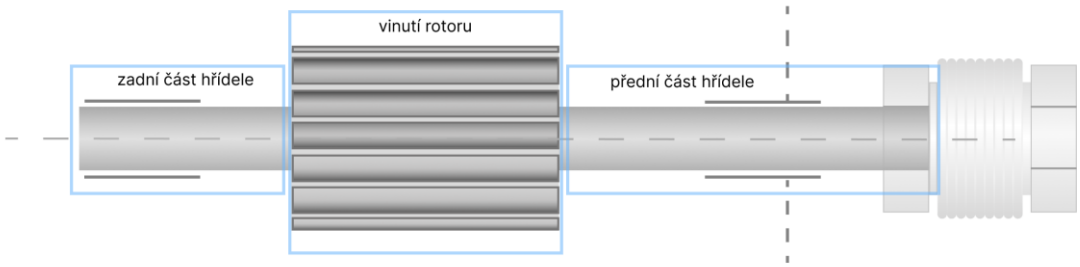
Název výstupu	Popis
connection	Třída s parametry spojení (setrvačnost, torzní tuhost)
shaft_changed	Aktualizovaná třída (zkráceného) hřídele – odebrány segmenty
hub	Aktualizovaná třída (zkrácené) armatury
shaft_end	Vytvořený nový prvek – nekonečně tuhá zbylá část hřídele

Výstupní třída *connection* je složena z vlastností *J* (moment setrvačnosti) a *tor_stf* (torzní tuhost), které mají vliv na dynamiku modelu. Ostatní vlastnosti třídy slouží k vizualizaci spoje. Parametry *J* a *tor_stf* jsou spočteny jako součty těchto parametrů zvolených segmentů označených indexem *idx* a *hubIdx*. Předtím je ale určen bod tuhosti. Označený segment u armatury není odstraněn, pouze zkrácen o vzdálenost po bod tuhosti. U hřídele je segment odebrán a nahrazen spojením a tuhou částí hřídele.

5.6 Modelování motoru

Projekt se soustředí na mechanické vlastnosti motoru. V modelu není zahrnuto řízení. Při vývoji se ale počítalo s možností rozšíření tohoto digitálního dvojčete o možnost řízení a využití modelu pro další analýzy.

Model motoru se v průběhu projektu měnil. Nebyl k dispozici žádný výkres, pouze katalogové informace. Citlivostní analýzy ale ukazovaly, že detailnější model motoru přináší přesnější výsledky při validaci. Během vývoje pak bylo experimentováno s různými konstrukcemi motoru. Model byl nakonec validován se strukturou rotoru rozdělenou do tří částí (viz obrázek. 28).

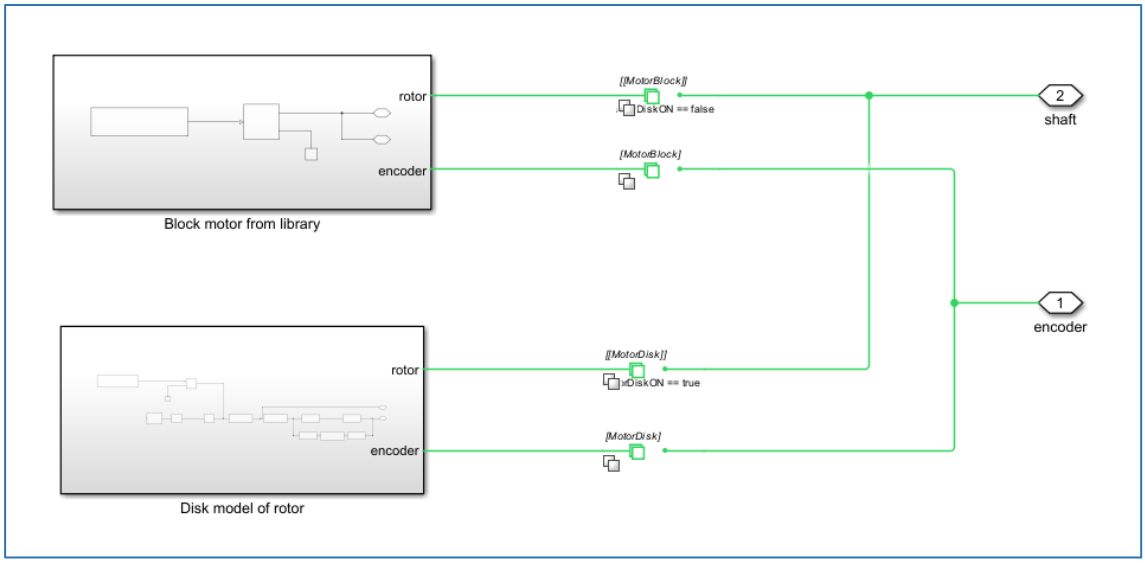


Obrázek 28 Rozdělní rotoru do tří částí

Rozdělení mechanických parametrů na rotoru bylo získáno po komunikaci s dodavatelem. Byl získán odhad torzní tuhosti rotoru. Při testech se ukázalo, že právě tato torzní tuhost má nemalý vliv na výsledné hodnoty rezonancí systému.

Během vývoje modelu byla kladena snaha na využití katalogových dat od výrobců jako základu. Proto s výše popsanou konečnou variantou bylo paralelně vyvíjeno další řešení, kde je motor reprezentován blokem *Machine Mechanical Power*. Výhodou oproti diskovému modelu je, že v modelu lze reprezentovat dynamiku motoru bez větší analýzy jeho konstrukce. Všechny parametry pro definování bloku lze získat z katalogu od

výrobce. V pozdějších fázích projektu se ale ukázalo, že diskový model je přesnější než motor reprezentovaný blokem *Machine Mechanical Power*. V modelu byly ponechány obě varianty pro případ, že by pro zadanou konfiguraci nebylo možné získat detailní parametry rotoru od výrobce. Varianty jsou přepínány pomocí bloků *Variant Connector* a BOOL proměnné *motorDiskON*.



Obrázek 29 Zapojení dvou variant motorů v subsystému Motor

Dále bude popsáno jen řešení s diskovým modelem. Vychází z celkového momentu setrvačnosti rozděleného na tři části: zadní hřídel, vinutí a přední hřídel, která je z části viditelná. Je také známa přibližná hodnota torzní tuhosti od středu vinutí až po konec přední hřídele. V katalogových informacích jsou uvedeny rozměry viditelné části přední hřídele, což umožňuje vymodelovat spojení motoru se spojkou.

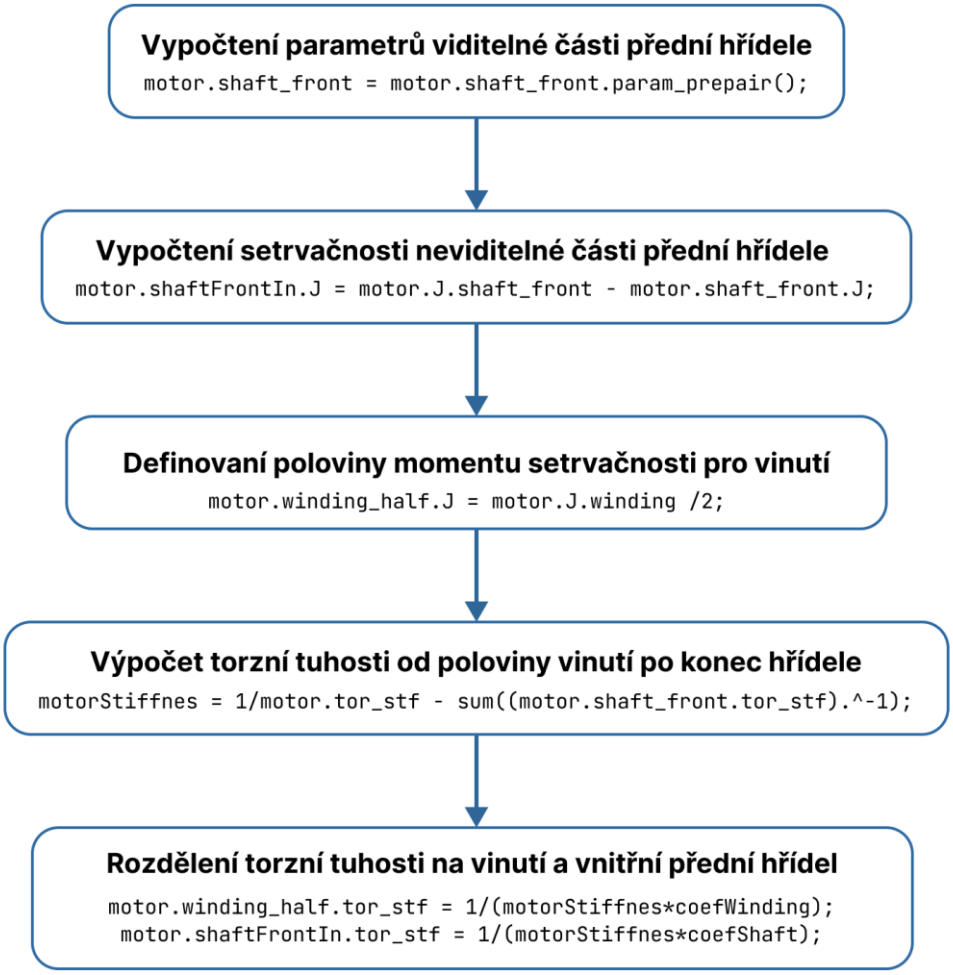
Materiálové vlastnosti viditelné části rotoru jsou uvažovány jako běžná ocel. Z určené hustoty, modulu pružnosti ve smyku a ze známých rozměrů viditelné části hřídele je vypočítána její setrvačnost a torzní tuhost. Spočtené parametry viditelné části jsou odečteny od parametrů definovaných pro celou část přední hřídele, která je tak rozdělena na dvě části.

V kódu jsou parametry modelu motoru uloženy ve třídě *motorClass* pojmenované jako *motor*. Hodnoty rozděleného momentu setrvačnosti na rotoru získané od výrobce jsou uloženy ve struktuře *motor.J* pod názvy *total* (celkový moment setrvačnosti motoru, který lze zjistit i v katalogu), *winding* (setrvačnost na vinutí), *shaft_front* (setrvačnost na přední hřídeli), *shaft_back* (setrvačnost na zadní hřídeli). Torzní tuhost známá od dodavatele je uložena v proměnné *motor.tor_stf*. Další vlastnosti třídy jsou popsány v tabulce 5.

Tabulka 5 Popis proměnných třídy motorClass

Název proměnné	Popis
J	Struktura setrvačností rozdělených na rotoru – vysvětleno v textu
tor_stf	Torzní tuhost rotoru od poloviny vinutí po konec předního hřídele
torque_constant	Konstanta z katalogu, využitá pro řízení motoru řídicím signálem
shaft_front	Třída *mechanicPart* – viditelná část přední hřídele
shaftFrontIn	Struktura obsahující *J* a *tor_stf* neviditelné části přední hřídele
winding_half	Struktura obsahující *J* a *tor_stf* poloviny vinutí
free	Moment setrvačnosti tuhé části přední hřídele (viz kapitola. 5.5)

Třída *motorClass* má dvě metody. Funkce *create()* přiřadí výchozí hodnoty parametrům a využívá se v GUI (anglicky graphical user interface – grafické uživatelské rozhraní, viz kapitola 5.13). Funkce *calculate(coefWinding, coefShaft)* vypočítá parametry pro detailněji rozdělený model. Pordrobnějšímu výpočtu koeficientů *coefWinding* a *coefShaft* se věnuje kapitola 5.10 o použití optimalizačního algoritmu pro identifikaci neznámých parametrů.



Obrázek 30 Algoritmus funkce motor.calculate()

U spojení motoru a spojky lze definovat, v jaké vzdálenosti od krytu motoru je spojka na hřídeli namontována. Pokud spojka není umístěna těsně u krytu motoru, jsou parametry viditelné přední hřídele *motor.shaft_front D, d* a *lng_sgmnt* rozděleny do dvou prvků pole. Tím se viditelná hřídel rozdělí dva segmenty. Vnější a vnitřní průměry jsou u obou segmentů stejné, délka prvního segmentu *motor.shaft_front.lng_sgmnt(1)* je rovna vzdálenosti od krytu motoru, ve které je spojka namontována. Pokud je spojka namontována těsně u motoru, skládá se viditelná hřídel pouze z jednoho segmentu.

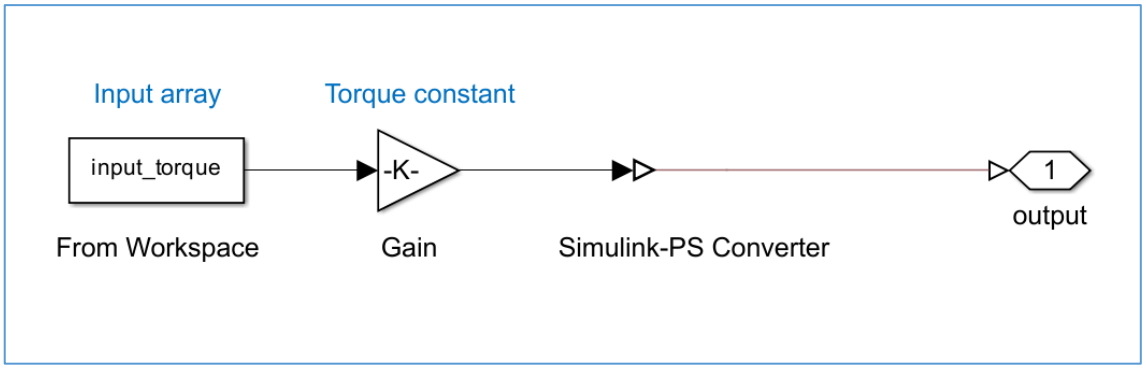
Pro modelování obou variant byly využity bloky variant *Variant Connector*, které jsou ovládány BOOL proměnnou *noDistanceMotor*, jejíž hodnota se automaticky určí na základě počtu segmentů hřídele *motor.shaft_front.n_segment.* Zapojení v Simscape s analogií ke schématu je zobrazeno v příloze II.

5.7 Budicí signál

V kapitole 2.3 byl v systému jeden z disků rozpohybován sinusoidálně se měnícím momentem a tím se rozkmital celý mechanický systém. Pro nalezení vlastních frekvencí je potřeba, aby byl systém těmito frekvencemi buzen. V praxi se jako budící signál používá bílý šum. Tento druh signálu má stejnou energii na všech frekvencích v požadovaném pásmu.[37]

U reálného systému popisovaného barevníku je kroutící moment osy generován v místě vinutí na rotoru. Proto i v modelu je systém buzen na rotoru, a to konkrétně v polovině vinutí. Od poloviny vinutí po konec rotoru je známá torzní tuhost rotoru (viz kapitola 5.6). V modelu je možné budit systém dvěma způsoby – vlastním signálem nebo signálem generovaným blokem *Band-Limited White Noise*.

Pro buzení systému vlastním signálem byl použit signál generovaný při frekvenční analýze na reálném stroji, který byl při měření uložen. Jedná se o proud generující kroutící moment, jehož hodnoty byly uloženy se vzorkovací periodou 0.25 ms. Pole hodnot bylo uloženo do proměnné *input_torque*, která je načtena z MATLAB Workspace do Simulink modelu blokem *From Workspace*. Hodnoty jsou násobeny konstantou kroutícího momentu, která je zjistitelná z katalogových informací u požitého typu motoru. Pro převedení Simulink signálu na signál fyzikální je použitý blok *Simulink-PS Converter* (viz obrázek 31).



Obrázek 31 Blokové zapojení pro buzení systému vlastním signálem

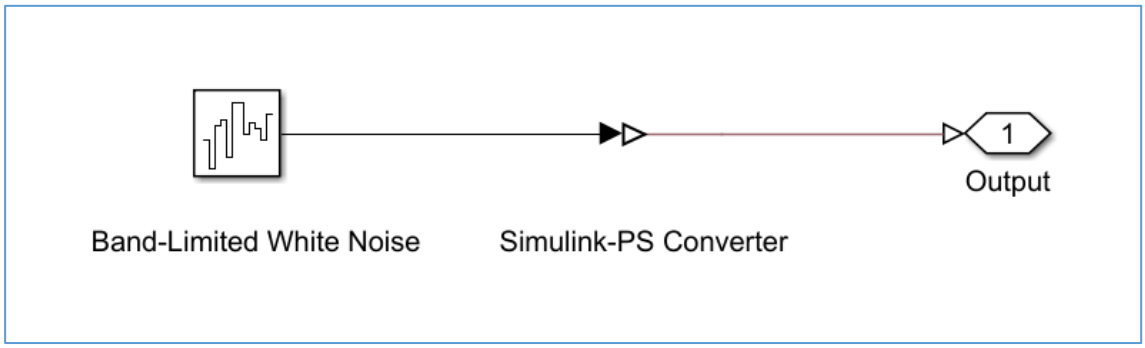
Tento způsob buzení byl při vývoji použit pouze pro porovnání se signálem generovaným blokem *Band-Limited White Noise*. Stejným přístupem by mohl být model rozšířen o možnost řízení motoru řídicím signálem.

U zapojení s blokem *Band-Limited White Noise* je potřeba nastavit vzorkovací periodu. Aby se zabránilo aliasingu, je vhodné nastavit vzorkovací frekvenci na dvojnásobek nejvyšší měřené frekvence. U bloku *Band-Limited White Noise* je doporučeno nastavit vzorkovací periodu podle:

$$tc = \frac{1}{100} \cdot \frac{2\pi}{f_{max}} \tag{5}$$

kde:

tc	=	Vzorkovací perioda	[s]
f_{max}	=	Nejvyšší frekvence systému	[rad/sec]



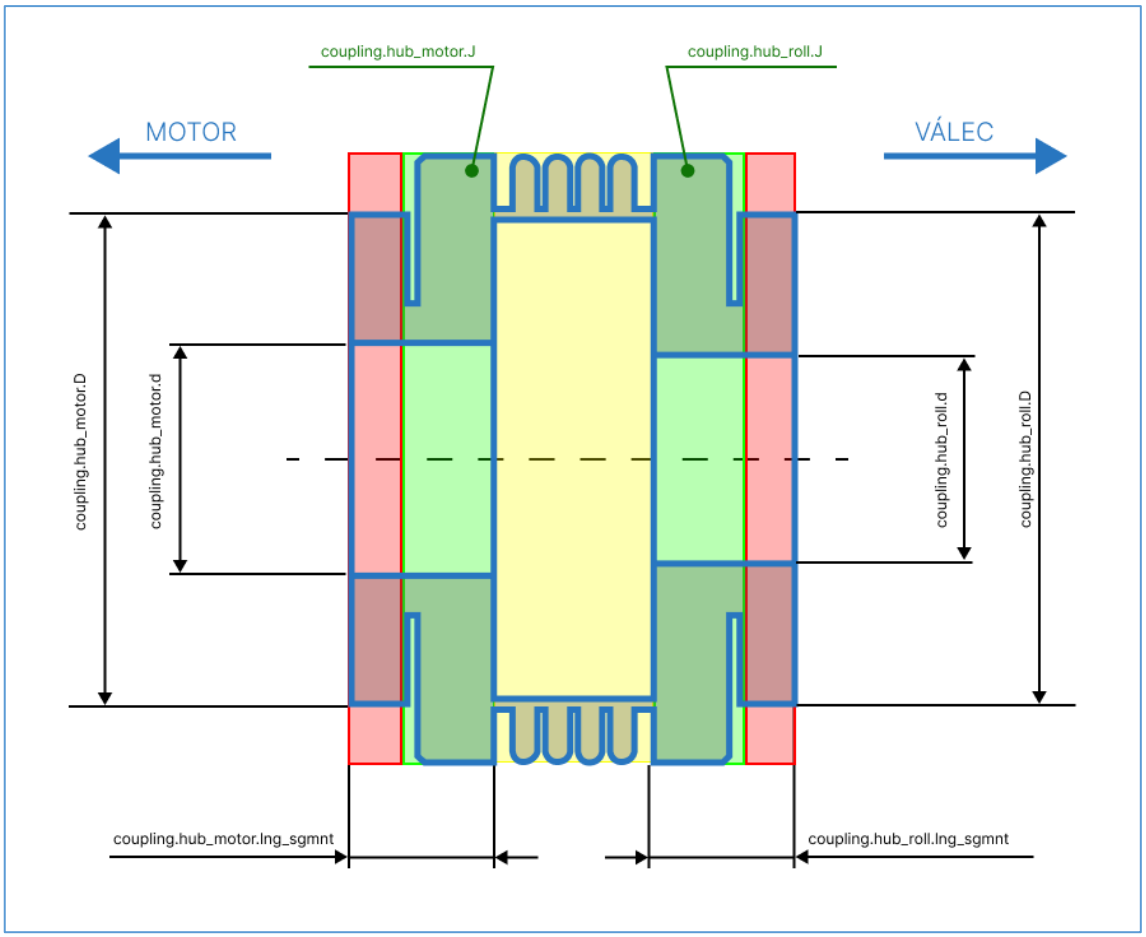
Obrázek 32 Blokové zapojení pro buzení blokem Band-Limited White Noise

Vzorec (5) a informace o bloku *Band-Limited White Noise* byly získány z MATLAB nápovědy [38]. Oba popisované způsoby zapojení jsou součástí subsystému *Source signal generator*, který lze vidět v příloze II.

5.8 Modelování spojky

Jednoduchý princip pro modelování spojky byl vysvětlen v kapitole 2.3, kde byla spojka v modelu reprezentována jako setrvačnost rozdělená na dva disky propojená torzní pružinou. Pro projekt byl tento zjednodušený přístup nedostačující, proto byl vyvinut detailnější model spojky. Opět byl při vývoji kladen důraz na to, aby všechny potřebné parametry byly dostupné z katalogových dat od výrobce.

Model spojky je rozdělen do tří částí – náboje a vlnovec. Pro určení vlastností nábojů je nutné definovat jejich materiálové vlastnosti (modul pružnosti ve smyku, hustota) a rozměry. U složitějších konstrukcí spojky je nutné najít vhodnou aproximaci, kdy jsou náboje převedeny na disky, jejichž momenty setrvačnosti a torzní tuhosti jsou relevantní pro frekvenční analýzu. Na vymodelované náboje je poté aplikován algoritmus pro vypočtení spojení hřídele s nábojem (viz kapitola 5.5)



Obrázek 33 Schéma spojky rozdělené do modelovaných částí s popisy

Na schématu obrázku 33 jsou náboje vyznačeny zelenou a červenou barvou. Červené oblasti nábojů znázorňují části spojky, které jsou použity pro modelování spojení hřídele s armaturou. Prostřední část spojky je vyznačena žlutou barvou a je definována torzí tuhostí a momentem setrvačnosti. Výpočet prostřední bude popsán níže.

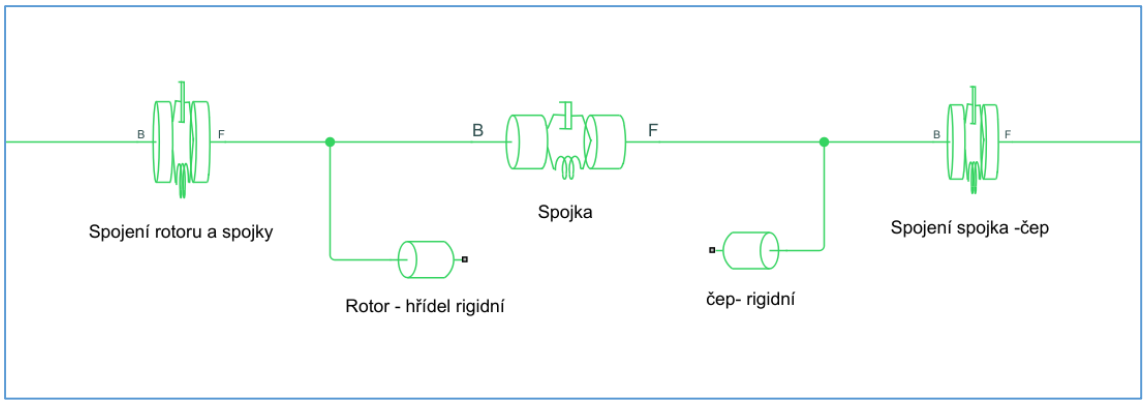
Parametry spojky jsou uloženy ve třídě *couplingClass* definované pod názvem coupling. Tato třída obsahuje čtyři vnořené třídy *mechanicPart*: *hub_motor* (náboj na straně motoru), *hub_roll* (náboj na straně válce), *mid* (vlnovec), *total* (celkové parametry pro spojku). Uložení prostřední části do třídy *mechanicPart* je zde kvůli vizualizaci. Jsou v ní uloženy i rozměry „vlnovce“, které nemají vliv na výslednou frekvenční analýzu.

Tabulka 6 Struktura třídy couplingClass

Název proměnné	Popis
hub_motor	*mechanicPart*, Náboj na straně motoru
hub_roll	*mechanicPart*, Náboj na straně válce
mid	*mechanicPart*, Prostřední část, důležité jsou parametry *J* a *tor_stf*
total	*mechanicPart*, Celkové vlastnosti spojky

Metoda *calculate()* počítá parametry pro prostřední část. Nejprve jsou spočteny setrvačnosti a torzní tuhosti nábojů metodou *param_prepair()*. Odečtením těchto hodnot od parametrů uložených ve třídě *total* jsou vypočteny hodnoty pro prostřední část *mid*.

Další metoda *vis()* vypočítá délku prostřední části *mid.lng_sgmnt* a přiřadí hodnoty průměrům. Vnější průměr vlnovce je stejný jako průměr náboje spojky. Vnitřní průměr je nastaven na nulu. Prostřední část slouží pouze pro vizualizační potřeby.



Obrázek 34 Blokové zapojení spojky v Simscape schématu

Spojka v zapojení na obrázku 34 je reprezentována blokem *Felxible shaft* se segmentací podle setrvačnosti a tuhosti. Zprava doleva je definována třídami *hub_motor, mid, hub_roll*. Jak již bylo zmíněno v dřívější kapitole, na pořadí definovaných prvků u bloku *Flexible shaft* záleží.

5.9 Volání simulace

Po zadání všech potřebných parametrů pro komponenty popisované v předchozích kapitolách je potřeba dopočítat zbylé parametry pro simulaci. Těmi jsou momenty setrvačnosti a torzní tuhosti jednotlivých segmentů a modelovaných spojení. V prvních fázích vývoje projektu byl pro tuto potřebu používán skript, ve kterém se provedly všechny potřebné operace. Pro různé typy sestavení systému byly změny prováděny ručně „zakomentováním“ nepotřebných částí (tím je myšleno například načítání souborů pro různé typy strojů). V pozdějších fázích se tento způsob stal nepraktickým, a proto byla vytvořena funkce *SimulationCall()*. Parametry funkce jsou popsány v následující tabulce:

Tabulka 7 Parametry funkce SimulationCall()

Název parametru	Popis
paramsFilePath	Cesta k souboru se vstupními daty
bridgeFilePath	Cesta k souboru s daty potřebnými pro simulaci s bridgem (volitelné)
inputParams	Změněné vstupní parametry (popsáno později)
simOutTime	Výstupní vzorky času simulace
simOutData	Výstupní vzorky měřených hodnot

V první fázi je načtený soubor s uloženými daty ve výše popisovaných třídách. Cesta je uložená ve vstupním parametru *paramsFilePath*. Poté je u načtených komponent aktualizován počet segmentů do proměnných *n_segment*.

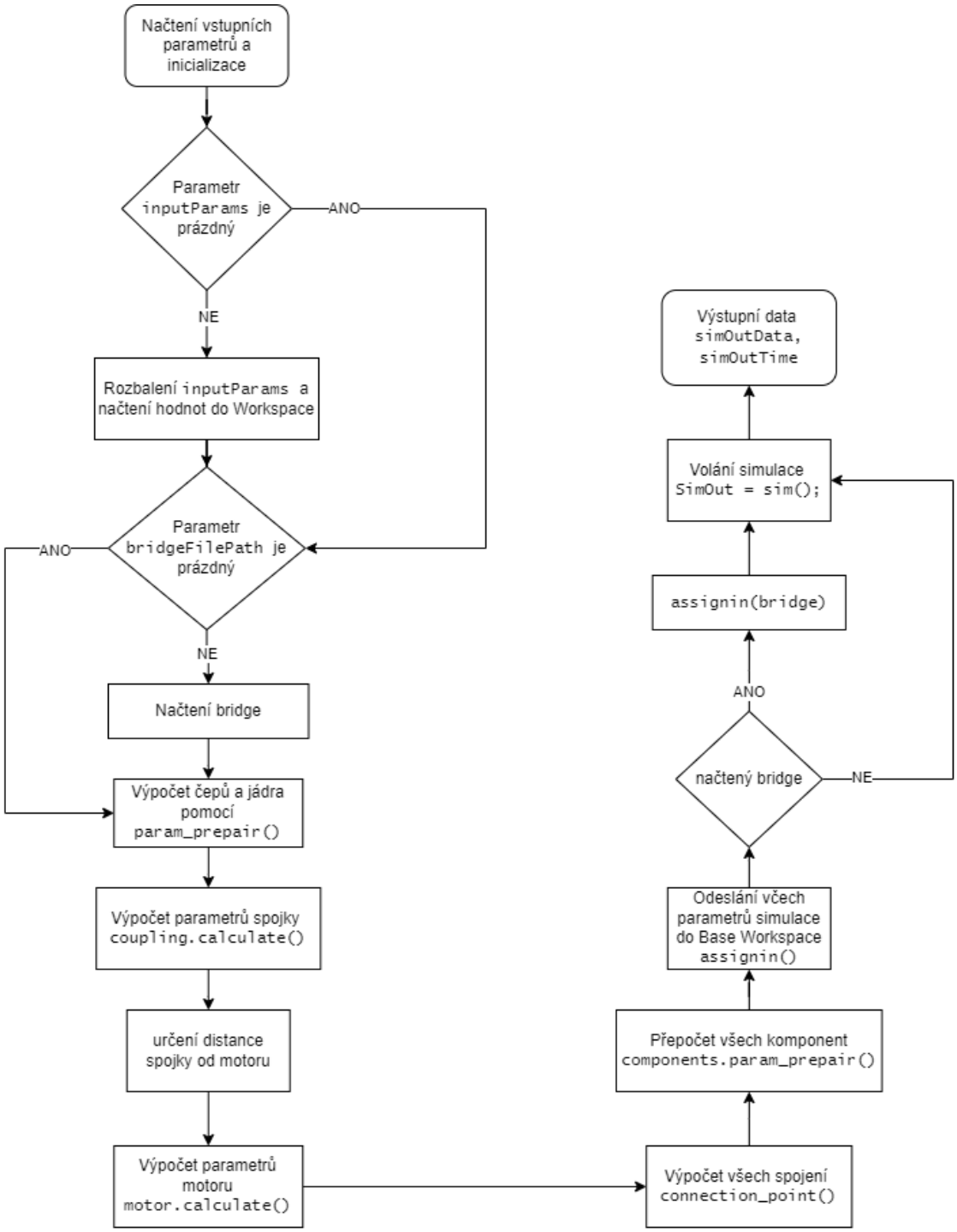
Funkce umožňuje při volání zadat parametry, které mají být změněny oproti uloženým parametrům v načítaném souboru. Toho bylo využito při ladění parametrů, které je popsáno v kapitole 5.10. Parametry se pouze přepíšou v MATLAB Workspace a není potřeba vyvářet další soubor obsahující všechny vstupní parametry. Takové načítání vstupů by bylo neefektivní.

Laděné parametry se zadávají jako buňkové pole v proměnné *inputParams*. Pro načtení parametrů do MATALB Workspace je použita funkce *eval()*. Funkce *SimulationCall()* akci provádí pouze v případě, že je proměnná *inputParams* neprázdná.

```
%% Unpacking input parameters
if exist('inputParams', 'var') ~= 0
    for i = 1:length(inputParams)
            paramInputName{i} = inputParams{1}{1}{i};
            paramInputValue{i} = inputParams{1}{2}(i);
            temp_name = paramInputName{i}{1};
            temp_value = paramInputValue{i};
            eval([temp_name '= temp_value']);
    end
    clear temp_value temp_name i;
end
```

Obrázek 35 Načítání laděných parametrů v kódu

V případě, že je neprázdná proměnná *bridgeFilePath*, provede se výpočet parametrů pro simulaci s bridgem. Podrobněji se tomuto problému věnuje kapitola 5.11. Následující algoritmus je popsán obrázkem 36.



Obrázek 36 Vývojový diagram funkce SimulationCall()

Podle diagramu jsou provedeny postupně výpočty potřebných parametrů podle postupů popisovaných v předchozích kapitolách. Po výpočtu spojení *connection* je nutné přepočítat momenty setrvačnosti a tuhosti u všech prvků systému, kde byly upravovány segmenty podle funkce *connection_point()*.

Všechny proměnné jsou uloženy v pracovním prostoru funkce, proto je nutné je sdílet s modelem v Simulinku. Pro tento účel byla použita MATLAB funkce *assignin()*, kde byl jako vstupní parametr nastaven základní pracovní prostor (Base Workspace), ze

kterého Simulink model dokáže načítat proměnné. Ve funkci jsou všechny názvy posílaných parametrů uloženy v proměnné *componentsWorkspace*. Hodnoty jsou jednotlivým proměnným přiřazovány ve *for* cyklu pomocí funkce *eval()*. Parametry pro bridge jsou posílány zvlášť pouze za podmínky, že má být v simulaci zahrnut. Simulace se spustí příkazem *sim()*.

5.10 Identifikace neznámých parametrů

U modelování je míra přesnosti modelu omezena dostupnými informacemi o parametrech reálného systému. Některé parametry jsou nedostupné anebo obtížně měřitelné. Existuje několik identifikačních metod, jak neznáme parametry odhadovat. MATLAB disponuje řadou toolboxů, které mohou být užitečné při řešení těchto odhadů. V této kapitole je navržen vlastní univerzální postup pro identifikaci a ladění parametrů.

Navrhovaný postup byl vyvinut na základě neznámých koeficientů pro rozdělení torzní tuhosti na hřídeli motoru (viz kapitola 5.6). Pro nalezení optimálního rozdělení byl využit genetický algoritmus z Global Optimization Toolbox. Ve vytvořené implementaci je pro výpočet účelové funkce (objective function) vyžadováno měření frekvenční analýzy z reálného stroje. Cílem bylo minimalizovat rozdíl mezi rezonancemi reálného stroje a modelu. Tento návrh byl volně inspirován článkem [39].

U hledání optimálního rozdělení torzní tuhosti na rotoru byly laděn jeden parametr *coefShaft*. Parametr *coefWinding* je na něm závislý. Je uvažováno že:

$$coefWinding + coefShaft = 1 \tag{6}$$

a torzní tuhost je rozdělena následovně:

$$tuhost_{vinutí} = \frac{1}{coefWinding \cdot tuhost_{motor}} \tag{7}$$

$$tuhost_{hřídel} = \frac{1}{coefShaft \cdot tuhost_{motor}} \tag{8}$$

kde $tuhost_{motor}$ je podle kapitoly 5.6 uložen v proměnné *motorStiffnes*.

Volba optimalizačního algoritmu

MATLAB Global Optimization Toolbox nabízí mnoho optimalizačních algoritmů. Cílem práce nebylo provádět srovnání jednotlivých algoritmů a hledání toho nejvhodnějšího, nýbrž vyvinout model tak, aby byla tato možnost dostupná. Reprezentativně byl vybrán genetický algoritmus.

Pro rychlost procesu byla nastavena velikost populace na 25 a počet generací na 15. Nebylo experimentováno s dalším nastavením, jako je typ selekce nebo mutace. Účelová funkce je nazvána jako *SimulationError()*. Meze jsou nastaveny v proměnných *lowBound* = 0.1 a *upBound* = 0.9. Počet parametrů *numOfParameters* je nastaven na 1.

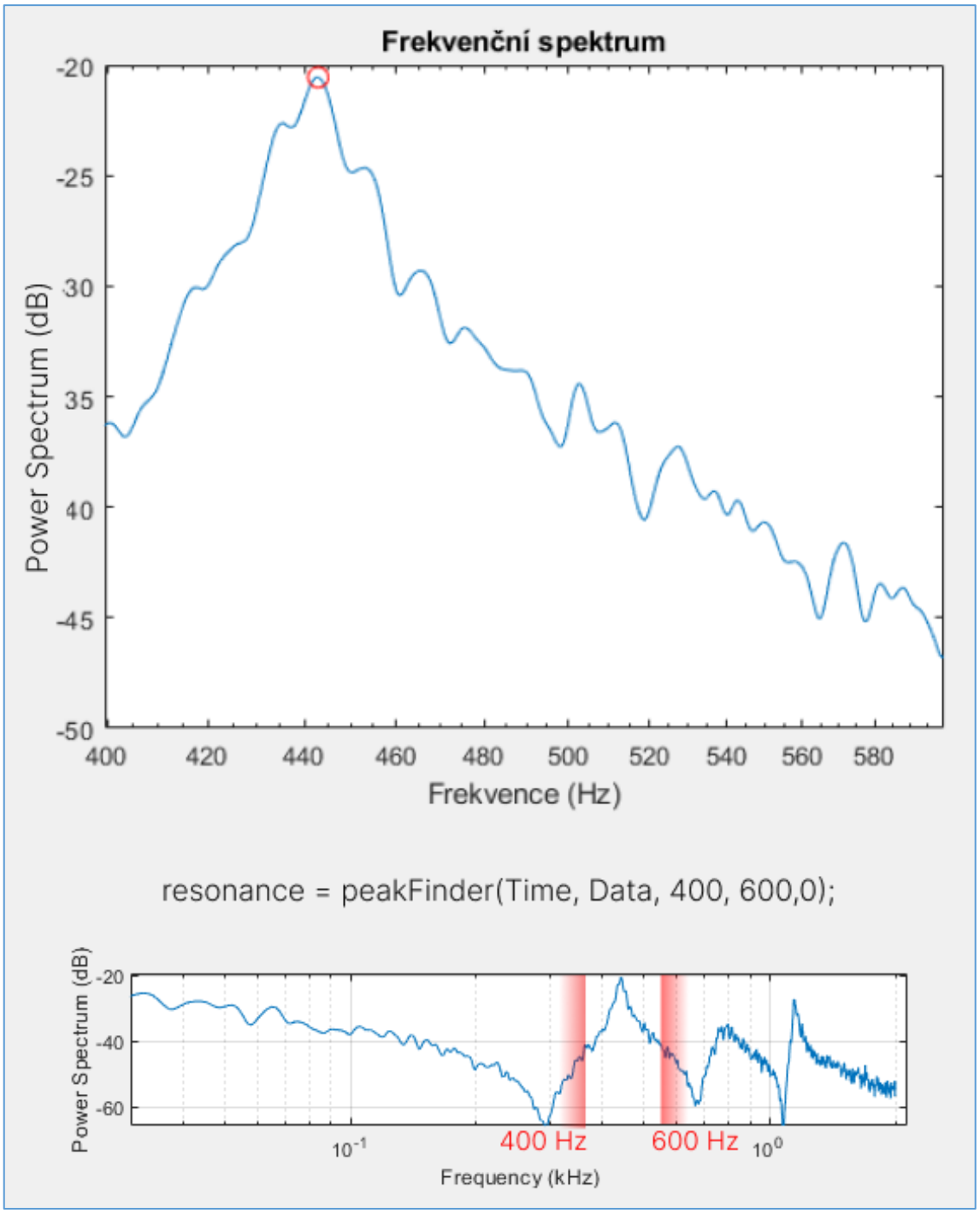
```
%% Calling Optimization Algorithm
% GA
algOptions = optimoptions("ga",'PopulationSize',25,'MaxGenerations',15,...
"Display","off");
% Solve
[solution,objectiveValue,~,output]=ga(@SimulationError, ...
numOfParameters,[],[],[],[],lowBound,upBound,[],[],algOptions);
```

Obrázek 37 Volání genetického algoritmu v kódu

Účelová funkce

Jak již bylo zmíněno, pro hledání optimálního nastavení parametrů je simulace porovnávána s frekvenční analýzou naměřenou na reálném stroji. Nevětší prioritou bylo zvýšit přesnost při odhadu první rezonance a antirezonance. Nižší priorita byla kladena na odhad druhé a třetí rezonance.

Pro nalezení rezonancí a antirezonancí v datech byla vytvořena vlastní funkce *peakFinder()*. Na vstupních datech provede frekvenční analýzu, ve které nalezne vrcholy charakteristiky. Funkce musí být volána pro každou rezonanci a antirezonanci zvlášť. Vstupními parametry funkce jsou i meze, ve kterých se má rezonance hledat. Funkce vrací hodnotu frekvence, na které byla nalezena rezonance (resp. vrchol charakteristiky). Hledání antirezonance je specifikováno při vlání funkce přidáním posledního BOOL parametru nastaveného na TRUE. Ve funkci *peakFinder()* jsou pak vstupní data invertována, aby byl ve spektru nalezen vrchol křivky. Meze je nutné odhadnout z celkové frekvenční analýzy.



Obrázek 38 Hledání rezonancí pomocí funkce peakFinder()

Tímto způsobem jsou nalezeny rezonance a antirezonance u měření na stroji a u výstupních měření simulace. Rezonance není nutné hledat funkcí *peakFinder()*, můžou být zadány i jiné hodnoty rezonancí, ke kterým se má optimalizační algoritmus přiblížit. Vyhodnocení rezonancí u provedené simulace musí být prováděno automaticky, proto byla tato funkce vyvinuta.

Pro vyhodnocení rozdílu mezi vstupními daty a měřením byla navržena následující funkce:

$$g = (r_1 - ri_1)^2 + (ar_1 - ari_1)^2 \cdot 0{,}7 + (r_2 - ri_2)^2 \cdot 0{,}5 + (r_3 - ri_3)^2 \cdot 0{,}2 \qquad (9)$$

kde:

r_n	=	N-tý řád rezonance simulace	[Hz]
ri_n	=	N-tý řád rezonance měření	[Hz]
ar_n	=	N-tý řád antirezonance simulace	[Hz]
ari_n	=	N-tý řád antirezonance měření	[Hz]
g	=	Funkční hodnota	

Optimalizační algoritmus se snaží minimalizovat hodnotu g. Rozdíly mezi vstupními daty a daty ze simulace jsou umocněny, čímž se dosáhne vyšší hodnoty g u velkých rozdílů. Rozdíly rezonancí jsou násobeny podle jejich priority.

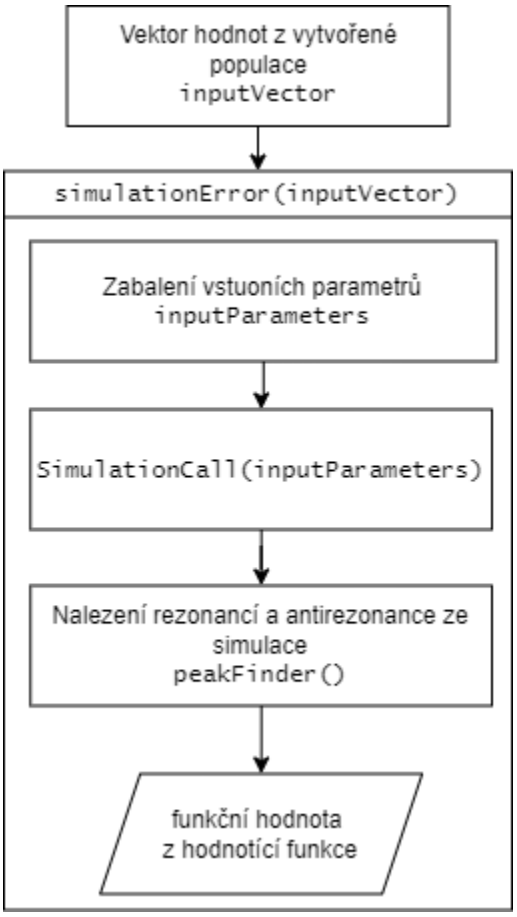
Spouštění algoritmu a volání simulace

Spouštění algoritmu bylo zahrnuto do funkce *optimizationOfParameters*, která vrací nalezené hodnoty hledaných parametrů. V následující tabulce jsou popsány proměnné pro nastavení optimalizace:

Tabulka 8 Proměnné funkce optimizationOfParameters()

Název proměnné	Popis
paramsToTune	Názvy parametrů pro ladění v modelu např. {“coefShaft”}
paramsFilePath	Cesta k souboru s daty načítanými do modelu
numOfParameters	Počet laděných parametrů
lowBound. upBound	Meze pro hledání hodnot parametrů
freqWindows	Okna pro hledání rezonancí a antirezonancí (matice)
solution	Nalezené hodnoty hledaných parametrů

Jako parametr *objective function* je genetickému algoritmu zadána funkce *SimulationError()*. Ta spouští simulaci s hodnotami parametrů z populace vytvořené genetickém algoritmem. Na obrázku 39 je popsáno, jak funkce *SimulationError()* vyhodnocuje úspěšnost vstupního vektoru.



Obrázek 39 algoritmus funkce SimulationError()

Při volání simulace je vhodné nastavit mód simulace na „accelerator“. Doba Simulace tak může být snížena, protože při nastavení velikosti populace na 100 jedinců a při provedení 25 generací se simulace provede 2500krát.

5.11 Rozšíření modelu o bridge

Při tisku větších obrázků se zvyšuje vnější průměr sleevu. S velkými raporty (délky forem se vzory) by musel být každý sleeve vyroben z velkého množství materiálu, což, by bylo náročné nejen z finančního hlediska, ale i z hlediska manipulace se sleevy. Řešením je nasadit na barevník nadstavec zvaný "bridge", na který lze pak nasunout tenký sleeve s větším raportem. Mezi zakázkami může bridge zůstat nasazený na barevníku, vymění se pouze sleeve.

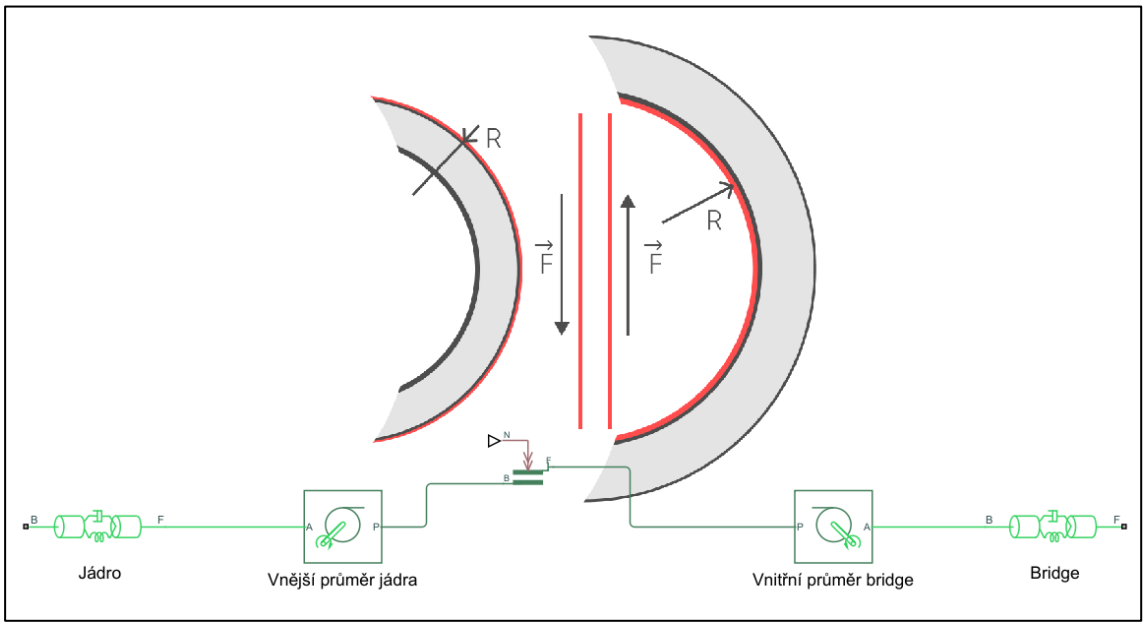
Kvalita bridge a sleevu má vliv na kvalitu tisku. Při měření na reálném stroji bylo prokázáno, že nasazený bridge mění dynamiku systému. To je motivací pro rozšíření původního modelu o simulaci s bridgem. Na obrázku 40 je schématicky znázorněno nasazení bridge na jádro.

Obrázek 40 Schéma jádra s nasazeným bridgem

Při nasazování je bridge roztažen stlačeným vzduchem, který proudí radiálními otvory jádra. Po umístění do správné polohy přestane stlačený vzduch proudit, bridge se stáhne a vznikne tlakové spojení.

Informace o materiálu bridge jsou velmi omezené, jelikož je dodáván třetí stranou. Při modelování byly známé pouze jeho rozměry a moment setrvačnosti. Pro validaci byla změřena frekvenční analýza na barevníku s nasazeným bridgem. Spoj ovšem nemůže být modelován stejně, jak tomu bylo u nasazení náboje na čep (kapitola 5.5), protože na reálném stroji může dojít ke stavu, kdy bridge po překročení určitého kroutícího momentu začne prokluzovat. Cílem bylo do modelovaného spojení mezi jádrem a bridgem zahrnout tření.

K přenosu kroutícího momentu dochází po obvodu jádra a tření je v modelování potřeba uvažovat jako translační pohyb, nikoliv rotační. Na následujícím blokovém schématu je naznačeno řešení, které bylo pro přenos kroutícího momentu modelováno:



Obrázek 41 přenos kroutícího momentu se zahrnutým třením

Pro převod mezi rotačním a translačním pohyb jsou použity bloky *Wheel and Axle*. Tření je zahrnuto v bloku *Loaded-Contact Translational Friction*. Na schématu 41 je přenášen pohyb mezi posledním diskem bloku *Flexible Shaft* jádra (port F) a prvním diskem bloku bridge (port B). V modelu bylo jádro s bridgem rozděleno tímto způsobem na 12 částí. Rozdělení do subsystémů je znázorněno v příloze III. Celá tato varianta jádra s bridgem je umístěna v subsystému pojmenovaném *kompozit – bridge*, který se aktivuje v závislosti na BOOL proměnné *bridgeON* (obrázek 26).

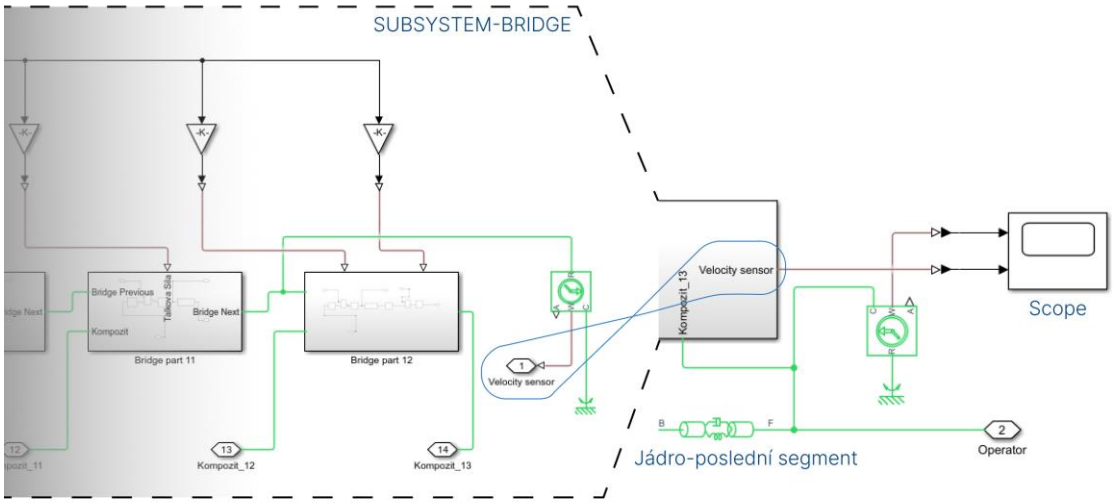
Rozložení síly po délce spojení je uvažováno nelineárně. Uprostřed tlaková síla působí nejvíce, naopak na krajích je nejmenší. V modelu je tlaková síla zadávána jako parametr. Bloků *Loaded-Contact Translational Friction* je použito 14. Aby bylo dosaženo efektu nižší tlakové síly, je síla před každým vstupem bloku *Loaded-Contact Translational Friction* násobena koeficientem (blok *Gain*). Uprostřed spojení je koeficient roven jedné, na kraji spojení je koeficient menší než jedna. Rozložení síly v modelu bylo modelováno podle:

$$f(x) = 1 - \frac{1}{550} \cdot (x - 7)^2. \tag{10}$$

Jedná se pouze o odhad. Model ale tímto způsobem umožňuje testovat i jiná rozložení tlakové síly.

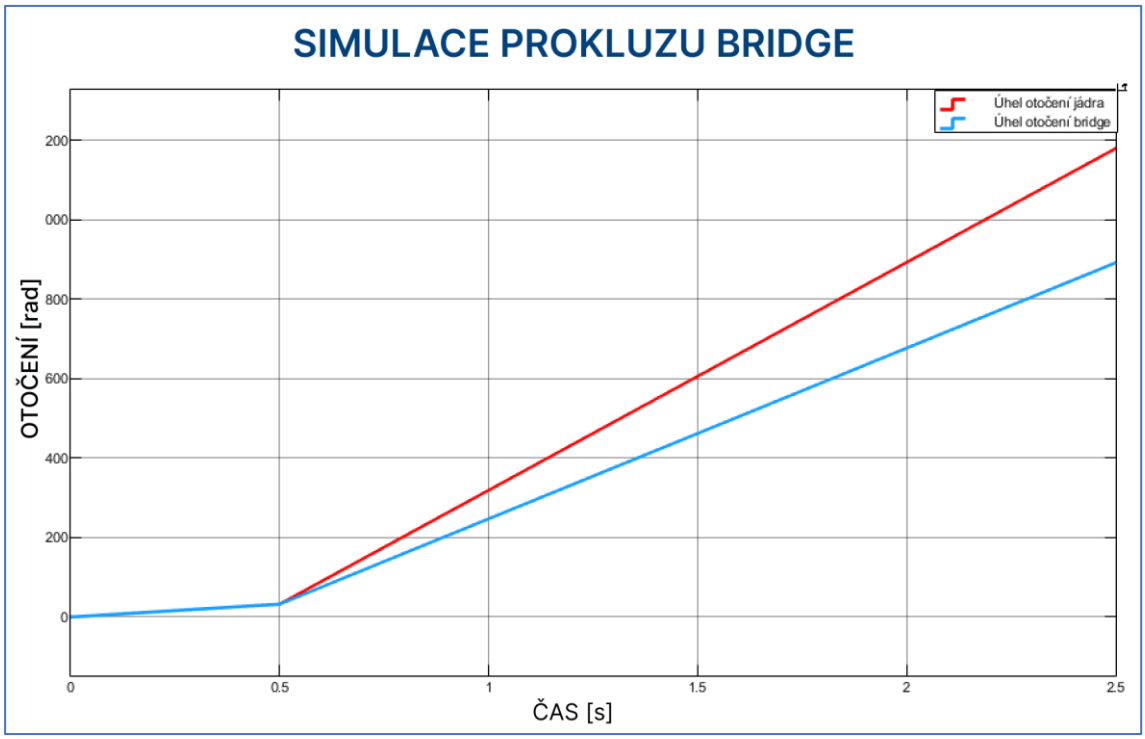
Vzhledem k tomu, že na bridge se ve výsledku opět nahlíží jako na diskový model, jsou parametry uloženy ve třídě *mechanicPart*. Jak již bylo zmíněno, jsou známy rozměry a moment setrvačnosti. Z těchto informací lze dopočítat váhu a hustotu. Modul pružnosti je uvažován do 1 GPa. Hodnoty parametrů definující tření mezi jádrem a bridgem jsou uloženy ve struktuře *bridgeSubsys*.

Verifikace modelu bridge spočívala v kontrole, jestli se nasazený bridge otáčí spolu s jádrem a zdali začne prokluzovat v případě překročení určitého kroutícího momentu. Pro tyto experimenty byl vstupní signál na rotoru (bílý šum) nahrazen blokem *Step*, který v definovaném čase skokově změní výchozí zadanou hodnotu signálu na jinou. Na následujícím obrázku je vyznačeno blokové zapojení pro měření otáčení:



Obrázek 42 Blokové schéma pro verifikaci

Na grafu níže (obrázek 43) je vyobrazeno porovnání otočení bridge. V čase 0.5 byla na vstup přivedena skoková změna, které měla za následek prokluz nasazeného bridge. Do času 0.5 se bridge otáčel stejně jako jádro.



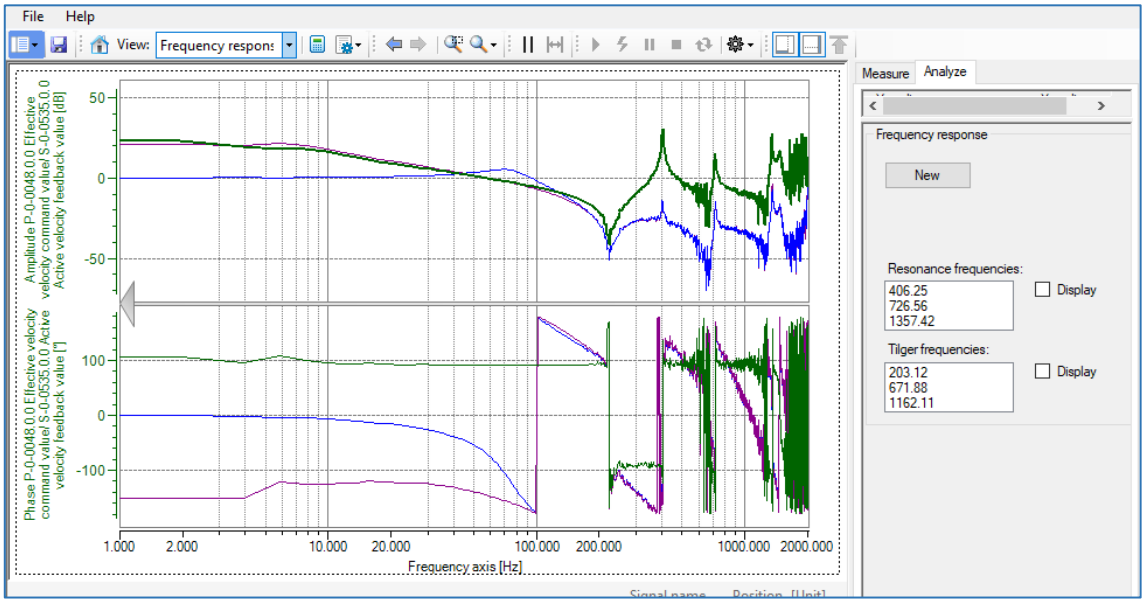
Obrázek 43 Výsledky verifikace bridge

Jedná se pouze o jeden z experimentů. Dále byl testován vliv přítlačné síly, koeficientů tření a různých typů řídícího signálu. Validace bridge je popsána v kapitole 5.12.

5.12 Validace modelu

Klíčovým krokem v procesu vývoje modelu je validace, jak již bylo zmíněno v kapitole 2.6 . Model byl vyvíjen pro stroj typu O20, z něhož byla získána data pro frekvenční analýzu. Předpokládá se, že všechny barevníky na stroji mají stejnou konstrukci a budou mít stejné vlastní frekvence. Nicméně, pro jistotu bylo provedeno měření na dvou různých barevnících. Pro měření byl použitý software IndraWorks od firmy Bosch, který umožňuje přímý přístup k datům z motoru v soustavě barevníku na stroji.

Výchozí měření bylo získáno pro tři varianty: tiskový válec bez sleevu, válec se sleevem (raport 500) a válec se sleevem (raport 800). Data jsou ukládána v souboru *.scope*, ze kterého je možné získat informace o vlastních frekvencích osy formového válce (obrázek 44). Pomocí softwaru IndraWorks DS byly ze souboru *.scope* exportovány sledované veličiny ve formátu *.txt*, který je možné importovat do softwarového prostředí MATLABu.



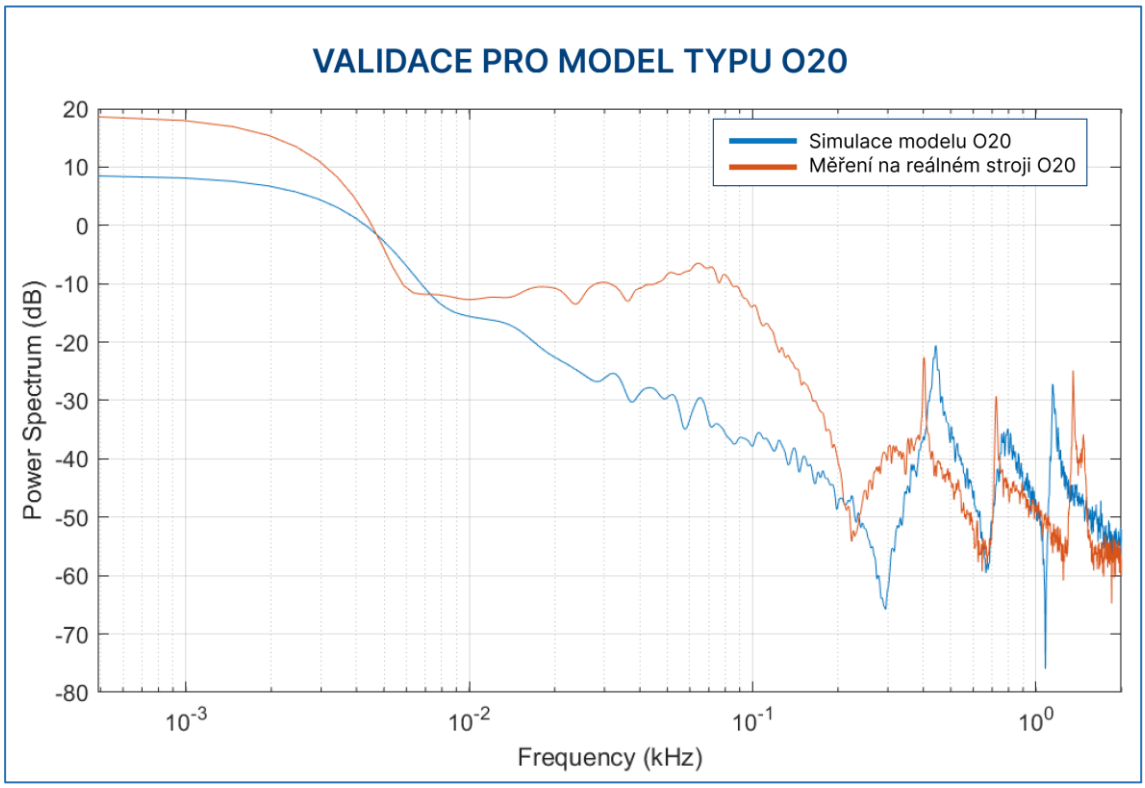
Obrázek 44 Frekvenční analýza zobrazená v IndraWorks DS - Oscilloscope

Pro každé měření byl také vygenerován soubor s parametry motoru, který obsahoval informace o aktuální zátěži (moment setrvačnosti osy), která se s nasazovaným materiálem na trubku měnila. Tento parametr byl později stěžejní pro určení materiálových vlastností bridge.

Sledovanými veličinami byly zpětná vazba rychlostní smyčky [rpm] a odpovídající časové vzorky se vzorkovací periodou 0,25 milisekundy. K dispozici bylo 8193 časových vzorků. Tyto veličiny byli vstupními parametry MATLAB funkce *pspectrum()*, která umožňuje analýzu signálu ve frekvenční doméně. Stejnou funkcí je analyzován výstupní signál ze simulace.

Porovnání výsledků ze simulace s měřením na stroji

Následuje porovnání modelu s reálným strojem pro typ O20:



Obrázek 45 Validace modelu typu O20

Tabulka 9 Výsledky validace modelu typu O20

	Model [Hz]	Reálný stroj [Hz]	Chyba modelu
První rezonance	442.9	402.4	10.0 %
První antirezonance	294.5	223.7	31.7 %
Druhá rezonance	796.1	723.3	10.0 %
Třetí rezonance	1144.3	1351.9	15.3 %

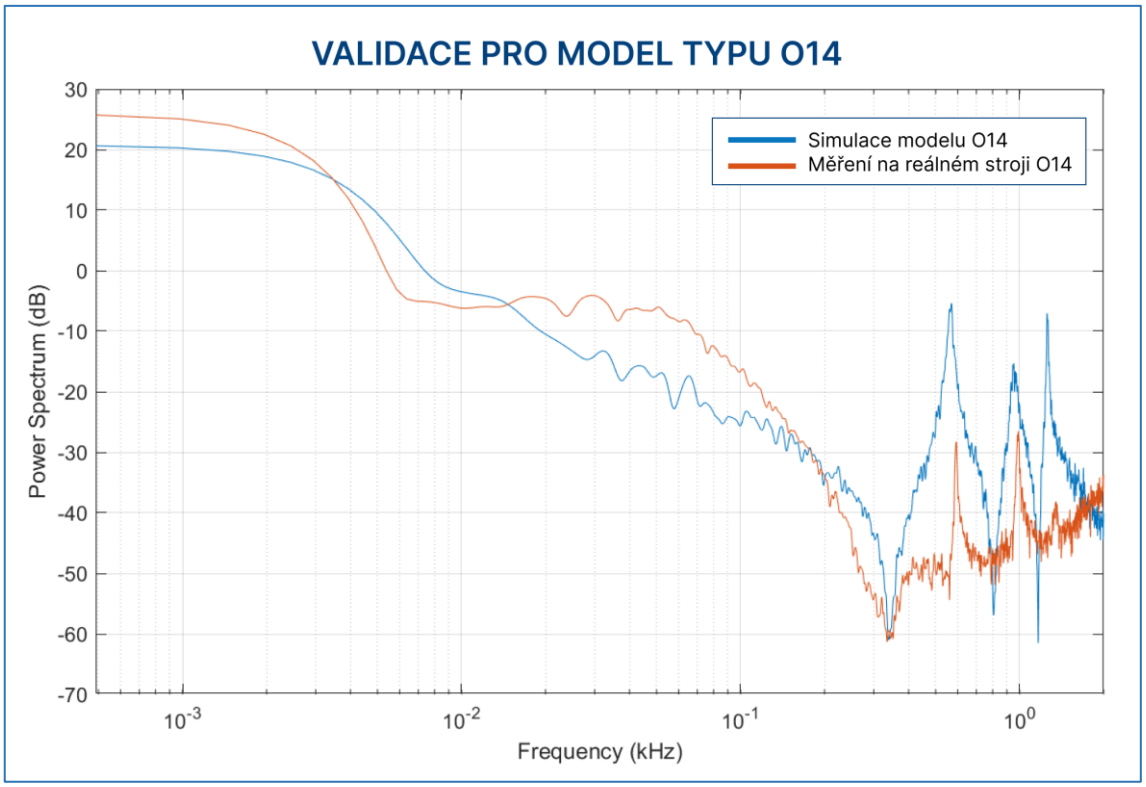
Relativní chyba modelu byla spočítána podle:

$$E_{rel} = \frac{f_{real} - f_{model}}{f_{real}} \cdot 100$$

kde:

E_{rel}	=	Relativní chyba modelu	[%]
f_{real}	=	Frekvence reálného systému	[Hz]
f_{model}	=	Frekvence modelu	[Hz]

Dále byl model validován se strojem typu O14. Získání dat z reálného stroje proběhlo stejným způsobem, jako bylo uvedeno dříve pro typ O20. Pro model byl vytvořen soubor se vstupními parametry stroje typu O14.



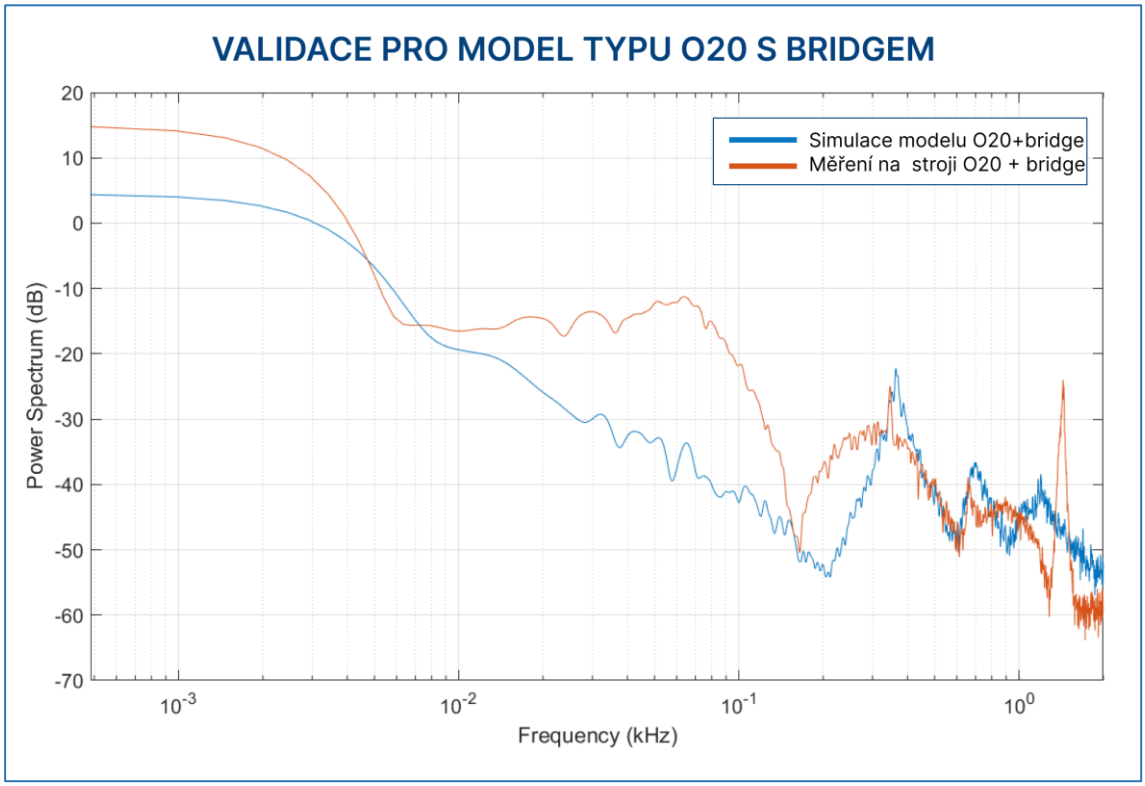
Obrázek 46 Validace modelu typu O14

Tabulka 10 Výsledky validace modelu typu O14

	Model [Hz]	Reálný stroj [Hz]	Chyba modelu
První rezonance	569.5	592.4	3.9 %
První antirezonance	339.9	335.0	1.5 %
Druhá rezonance	952.4	989.5	3.75 %
Třetí rezonance	1254.7	1346.5	6.8 %

Chyba modelu byla počítána stejně jako u experimentu pro typ O20.

Rozšíření modelu o nasazený bridge bylo validováno se strojem typu O20.



Obrázek 47 Validace modelu typu O20 s nasazeným bridgem

Tabulka 11 Výsledky validace modelu typu O20 s nasazeným bridgem

	Model [Hz]	Reálný stroj [Hz]	Chyba modelu
První rezonance	363.4	346.8	4.8 %
První antirezonance	212.0	165.0	28.4 %
Druhá rezonance	695.0	657.4	5.7 %
Třetí rezonance	1197.1	1436.9	16.7 %

Zhodnocení dílčích validací je komentováno v závěru práce.

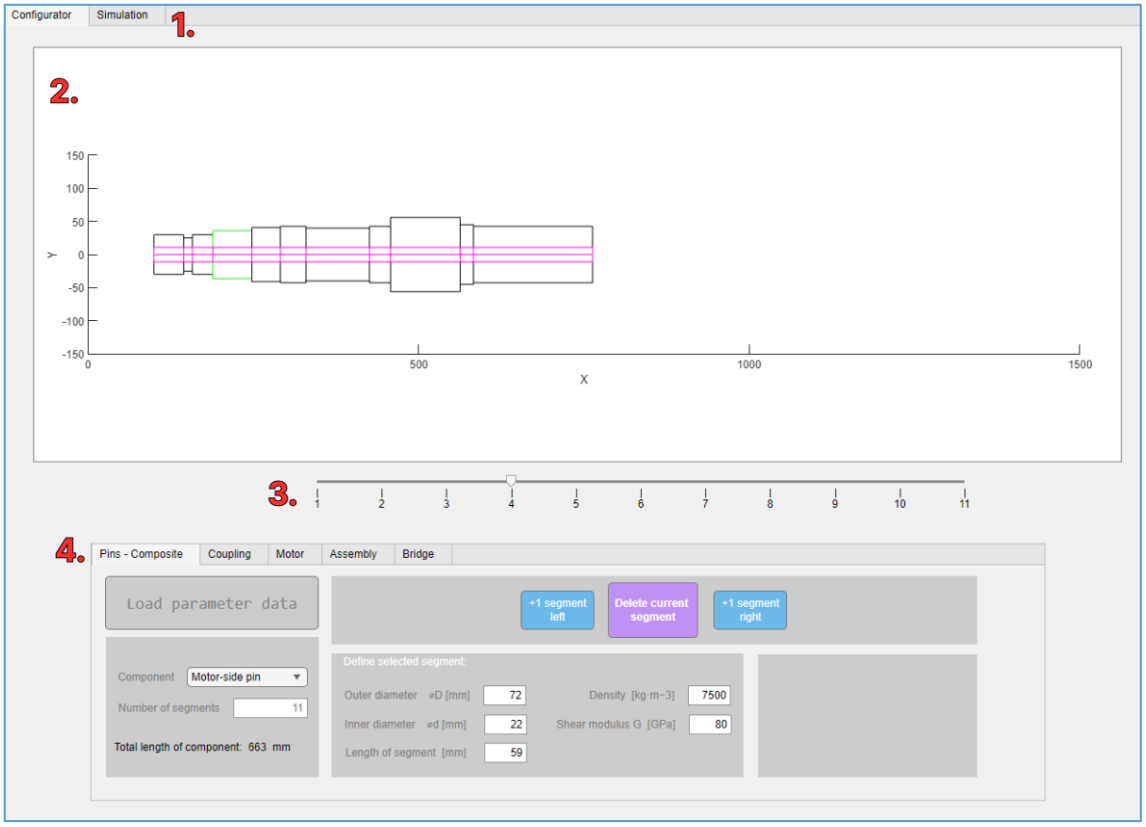
5.13 Uživatelské rozhraní

Popisovaný model je závislý na souboru se vstupními daty, která pak zpracovává. V dřívějších kapitolách bylo detailně popsáno, jak jsou data do modelu načítána. Uživatel modelu by v případě vytvoření nebo úpravy souboru dat musel mít znalosti se zadáváním příkazů v konzoli MATLAB a musel by umět pracovat se třídami, strukturami a poli. Z tohoto důvodu bylo vyvinuto uživatelské rozhraní, které umožní uživateli navrhnout nové komponenty, upravit již vytvořenou sestavu, testovat provedené změny a porovnávat výsledky simulace.

Pro vývoj GUI byl použit nástroj App Designer od MATABu. Cílem bylo vytvořit prostředí, které bude přehledné a jednoduše ovladatelné. Kvůli časové náročnosti vývoje nebyl kladen důraz na robustnost a aplikace nebyla vyvinuta s ohledem na responzivní design.

Aplikace byla v nástroji App Designer exportována jako MATLAB App. Uživatel si tedy vytvořené GUI nainstaluje přímo jako aplikaci v MATLABu.

Na následujícím obrázku jsou popsány oblasti GUI pro základní navigaci v okně aplikace. Název prvku „jádro“ je v GUI zaměněn za „composite“.

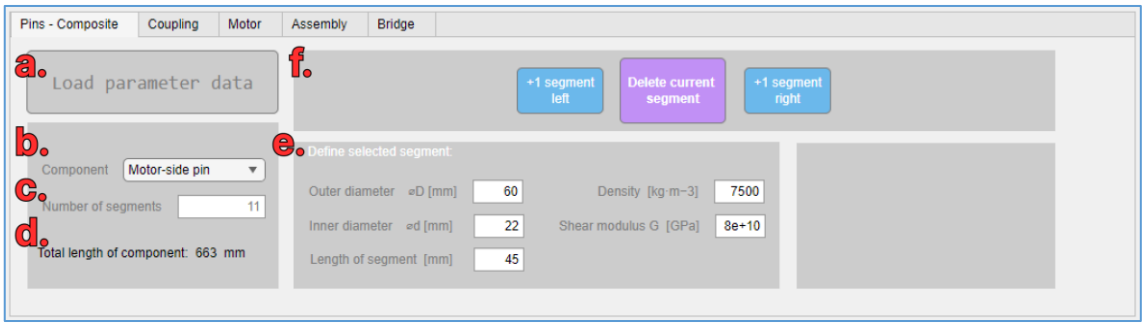


Obrázek 48 GUI – úvodní popis

Popis oblastí:

1. Záložky **konfigurace** a **simulace**
   V okně **konfigurace** jsou designovány jednotlivé komponenty a sestava. Lze zde načíst a ukládat soubory dat. Okno **simulace** bude popsáno později.
2. **Vizualizace komponent**
   Tato oblast zobrazuje aktuálně navrhovanou komponentu, aby uživatel měl okamžitou zpětnou vazbu o jejích rozměrech. Pro vizualizaci byla napsána vlastní funkce, která bude popsána později.
3. **Volba segmentu**
   Posuvník je viditelný pouze u návrhu čepů a jádra. Pokud se komponenta skládá pouze z jednoho prvku, je posuvník skryt. Aktuálně zvolený segment je ve vizualizaci (2.) zvýrazněn zelenou barvou.
4. **Konfigurace komponent**
   Oblast se skládá z pěti záložek: Pins-Composite, Coupling, Motor, Assembly, Bridge. Ve zvoleném okně je možné nastavit parametry pro zvolenou komponentu.

Pro návrh čepů a jádra je nutné mít aktivní záložku *Pins-Composite* v oblasti 4. Na následujícím obrázku bude popsáno rozhraní této záložky. Vzhledem k tomu, že jádro i čepy jsou v programu definovány stejnou třídou, je možné jejich parametry zadávat stejným způsobem.



Obrázek 49 GUI – rozhraní návrhu čepu

Popis oblastí:

a. **Načtení parametrů**
   Stisknutím tlačítka uživatel načte již vytvořený soubor parametrů.
b. **Volba komponent**
   Uživatel zvolí, jestli chce editovat čep u motoru, jádro nebo čep u operátora.
c. **Počet segmentů**
   Uživatel má možnost rychle upravit segmentaci zvoleného prvku. V případě zvolení vyššího čísla, než je aktuální, jsou přidány primitivní segmenty o velikosti 50x50 mm. V opačném případě jsou segmenty odebrány zprava.

d. Celková délka prvku
Zobrazovaná hodnota je součtem délek všech segmentů prvku. Uživatel má možnost rychlé kontroly s výkresem.

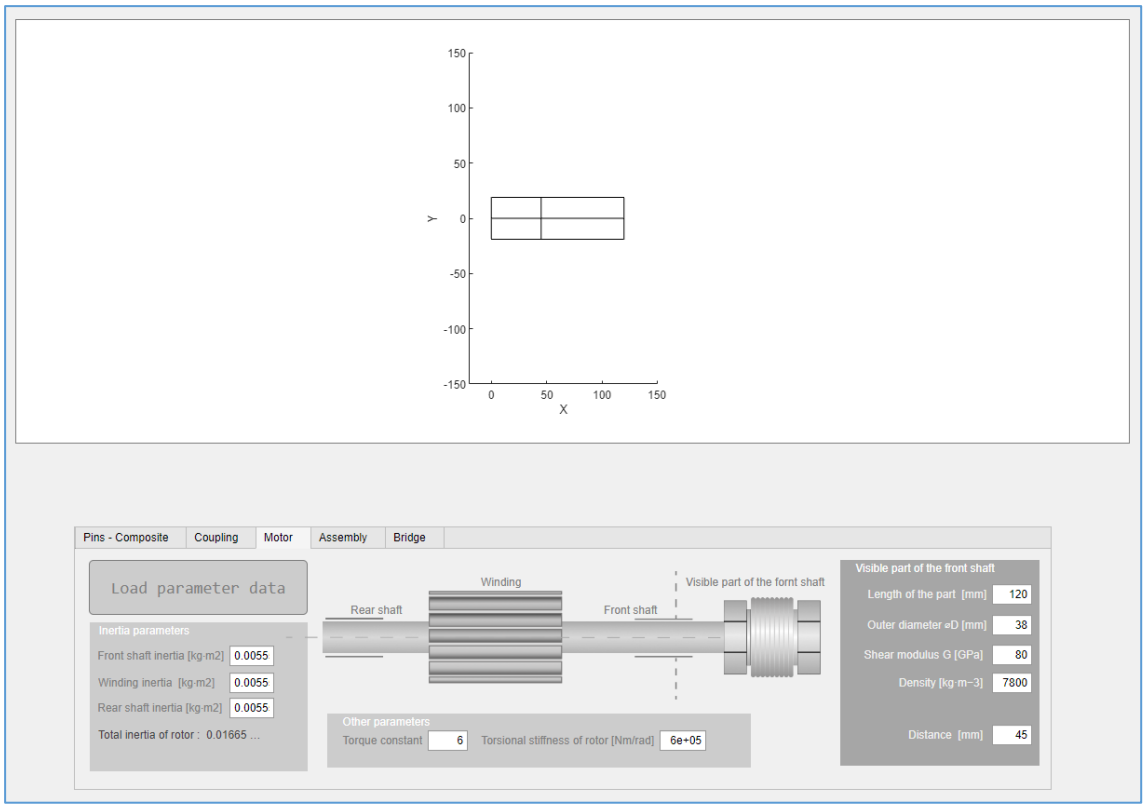
e. Editace parametrů
V této oblasti uživatel zadává hodnoty parametrů pro jednotlivé segmenty. Hustota a modul pružnosti ve střihu jsou pro všechny segmenty stejné.

f. Přidání/odebrání segmentu
Modrá tlačítka přidávají primitivní segment nalevo nebo napravo od zvoleného segmentu. Fialové tlačítko odebere aktuálně zvolený segment.

Při navrhování čepu u motoru musí být segmenty řazeny tak, aby vlevo byl segment spojující čep se spojkou. Naopak u čepu u operátora musí být segmenty řazeny tak, aby vlevo začínal segmentem spojeným s jádrem.

Na následujících snímcích jsou vyobrazeny záložky **Coupling** a **Motor** pro návrh spojky a motoru.

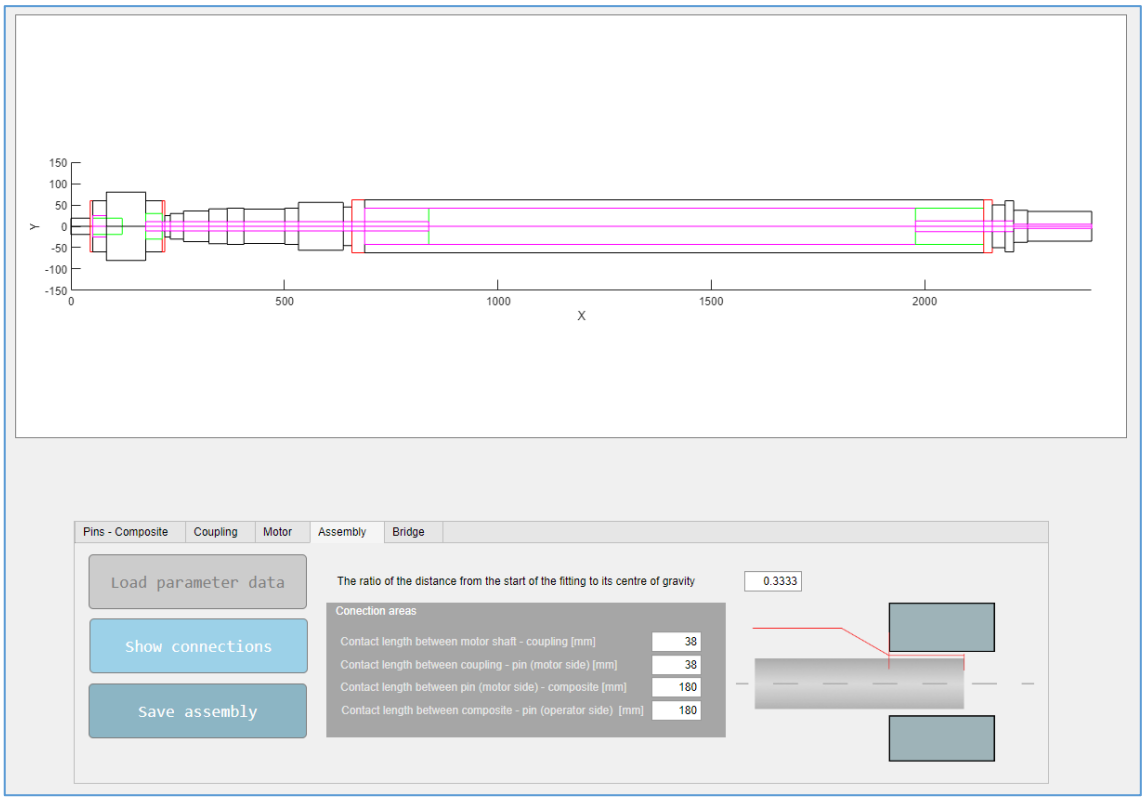
Obrázek 50 GUI – návrh spojky



Obrázek 51 GUI – návrh motoru

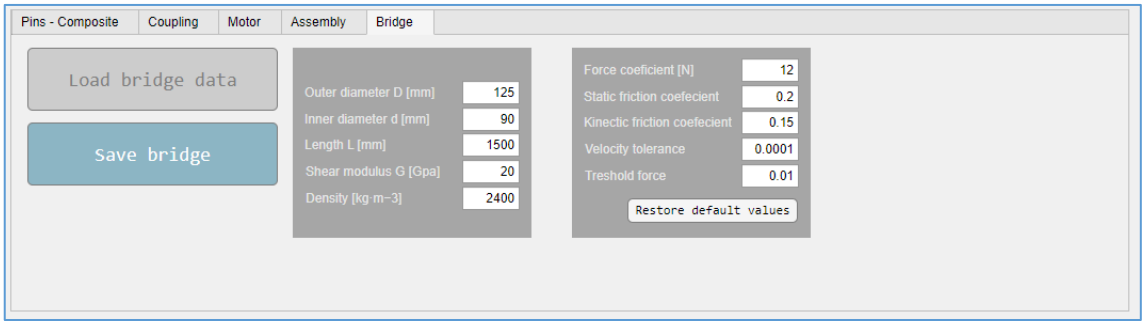
U motoru je zobrazována pouze viditelná část hřídele. Pokud je hodnota v poli *Distance* vyšší než nula, je hřídel rozdělena na dva segmenty. Hodnoty parametrů na snímcích z GUI jsou fiktivní.

V záložce **Assembly** jsou definovány parametry pro spojení mezi komponenty. Tlačítkem **Show connections** se zobrazí celá sestava v případě, že je správně navržena. Tlačítkem **Save assembly** lze uložit navržené parametry sestavy.



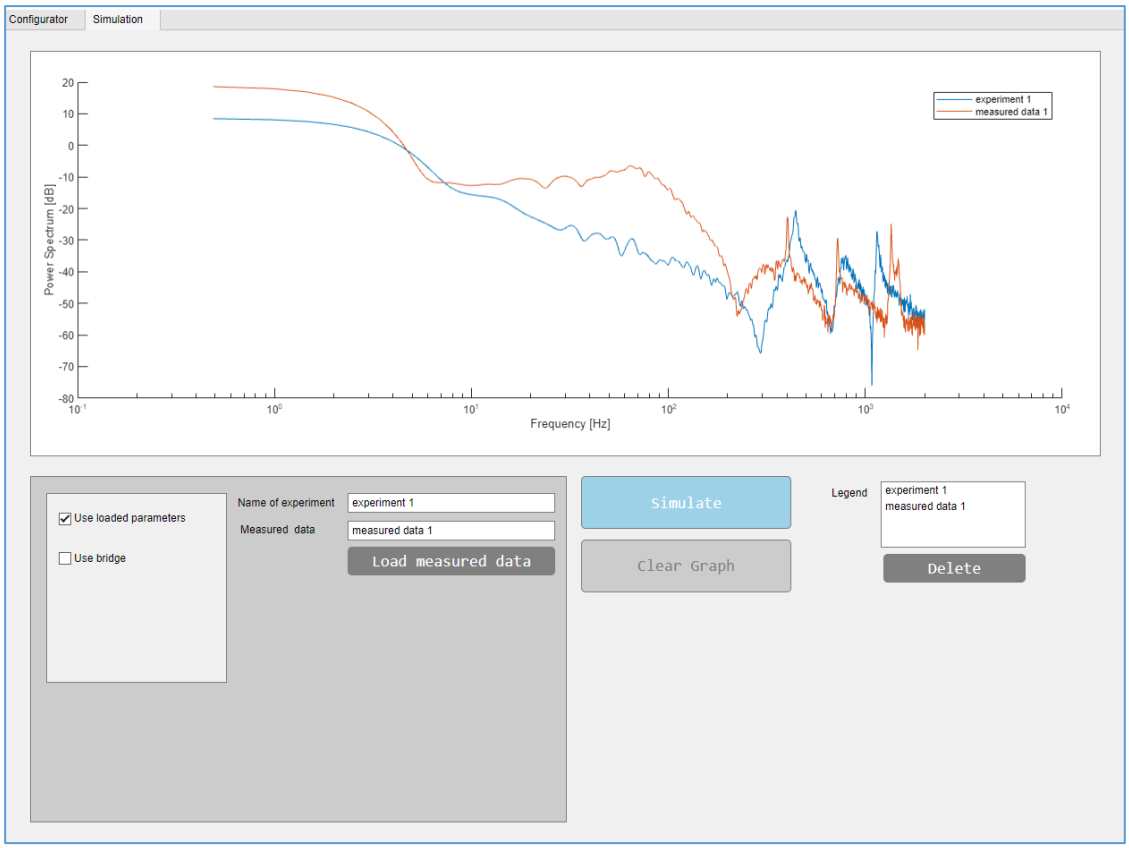
Obrázek 52 GUI – sestava

V záložce **Bridge** uživatel definuje parametry pro bridge a tření mezi bridgem a jádrem. Bridge je ukládán a načítán odděleně od celé sestavy. Ve vizualizaci se nezobrazuje.



Obrázek 53 GUI – bridge

V horní záložce **Simulation** má uživatel možnost testovat navržené parametry simulací modelu. Lze porovnávat mezi sebou výsledky experimentů s naměřenými daty. Tlačítkem **Simulate** uživatel spustí simulaci modelu s načtenými parametry. Volbou **Clear Graph** smaže všechny data z grafu. V oblasti nazvané **Legend** (nikoliv legenda v grafu) je možné vybrat křivku, která má být z grafu smazána. Naměřená data ze stroje se načtou tlačítkem **Load measured data**.



Obrázek 54 GUI – simulace

Náhled do kódu

V následujícím textu bude stručně nastíněno, jak GUI funguje na pozadí. V aplikace využívá stejných názvů atributů jako jsou komponenty v modelu (např *pinMotor*). Při spuštění aplikace se tyto atributy inicializují jako požadovaná třída (např. *mechanicPart*) a jsou jim přiděleny výchozí hodnoty. Například čepy jsou inicializovány jako primitivní segment s vnějším průměrem a délkou rovnou 50 mm.

Všechna pole, do kterých jsou zadávány parametry, jsou „namapována“ na příslušnou komponentu funkcemi „ValueChanged“. Pokud se hodnota v poli změní, zavolá se příslušná funkce (např. OuterdiameterDmmEditFieldValueChanged) která novou hodnotu přepíše v požadované třídě. Aplikace přitom bere v úvahu, jaká komponenta je aktuálně zobrazována a jaký segment komponenty je zvolen. Nástroj App Designer generuje názvy funkcí automaticky.

```
% Value changed function: OuterdiameterDmmEditField
function OuterdiameterDmmEditFieldValueChanged(app, event)
    value = app.OuterdiameterDmmEditField.Value;    % hodnota v poli D
    sliderIndex = app.SelectedsegmentSlider.Value;  % zvolený segment na posuvníku
    % zvolená komponenta v dropdown menu (např pinMotor)
    componentName = app.componentNames{app.ComponentDropDown.Value};
    % funkce eval přiřadí hodnotu v poli do třídy (pinMotor.D(sliderIndex))
    eval([componentName '.D(sliderIndex)'  '= value;']);
    app.componentPreview();     %aktualizace vizualizace


end
```

Obrázek 55 Zápis parametru do třídy

Při ukládání sestavy se do soubor s daty uloží následující atributy s jejich aktuálními hodnotami: motor, pinMotor, pinOperator, composite, coupling, contact_area. Jedná se o třídy a struktury, které jsou potřebné pro zavolání funkce *SimulationCall()*. Načítání souboru probíhá na podobném principu – načte se uložený soubor a hodnoty se nahrají do příslušných tříd. Soubor má příponu „.mat“.

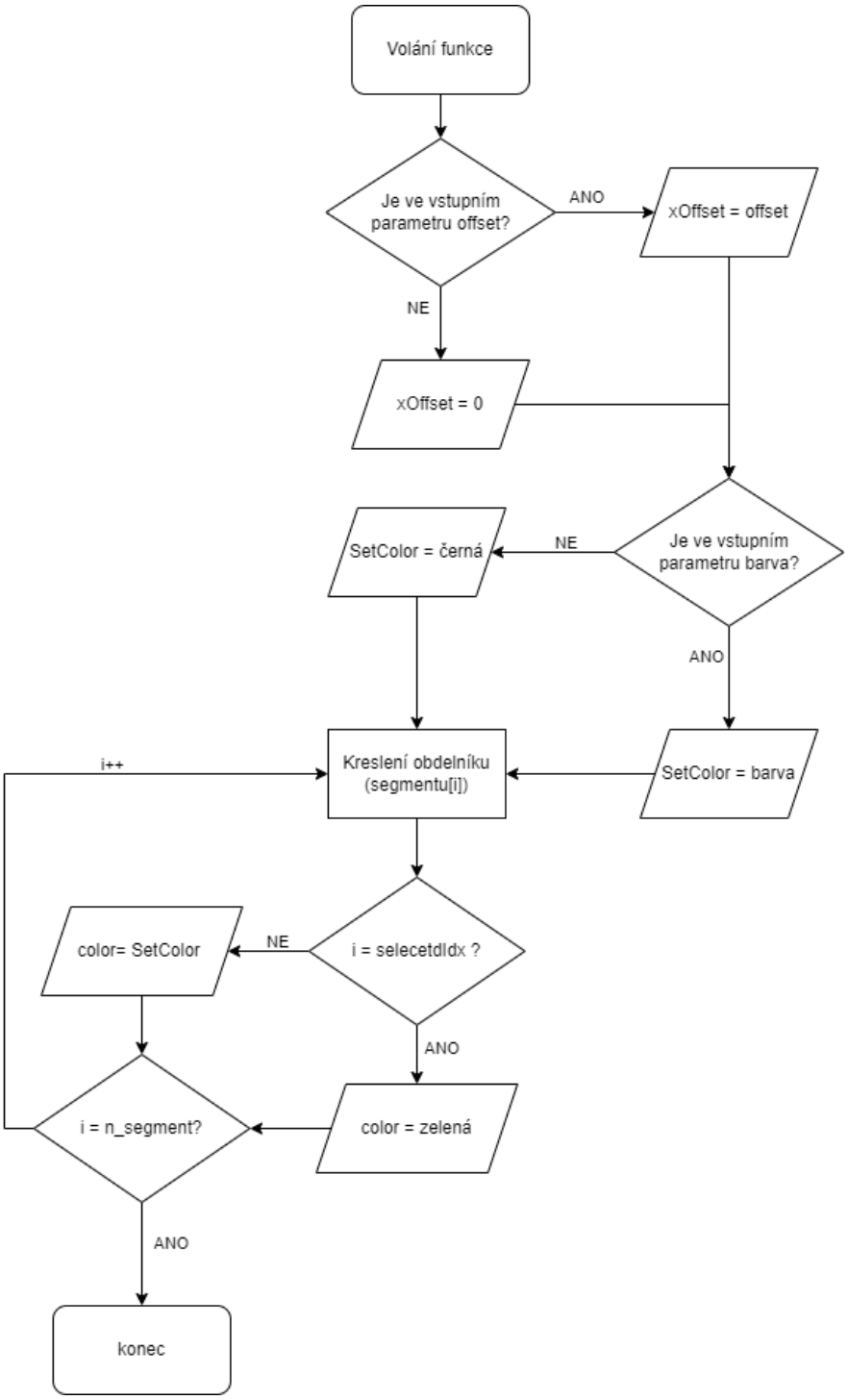
Simulace z uživatelského prostředí může být zahájena dvěma způsoby. Pokud uživatel zvolí možnost „Use loaded parameters“, použijí se při simulaci aktuální hodnoty parametrů v GUI. Simulace je spuštěna zavoláním funkce *SimulationCall()* jejímž vstupním parametrem je soubor s uloženými hodnotami parametrů. Je vytvořen dočasný soubor s aktuálními daty ve složce 'C:\Users\guest\AppData\Local\Temp\'. Stejným způsobem proběhne i uložení soboru s parametry pro bridge.

V případě, že při simulaci není zvolena možnost „Use loaded parameters“, otevře se uživateli okno s výběrem souboru, kde jsou uloženy parametry pro simulaci. Pokud chce uživatel provést simulaci s bridgem, musí být vždy načtený v GUI. Po provedení úspěšné simulace se zobrazí frekvenční analýza v grafu.

Jakékoliv změny rozměrů komponent se ihned vykreslí do vizualizace voláním funkce *componentPreview()*.

Funkce componentPreview a assebmlyPreview

Volání funkce *componentPreview()* probíhá ve většině funkcí typu „ValueChanged“, tedy při změně rozměrů ale i při zvolení jiné záložky komponenty (např. přechod ze záložky Pins-Composite na Coupling). Funkce funguje na vlastní vyvinuté funkci *visualization()*, která vykreslí komponentu zadanou jako vstupní atribut. Na následujícím obrázku je uveden stručný vývojový diagram pro funkci *visualization()*.



Obrázek 56 Vývojový diagram funkce visualization()

Dále je uvedena tabulka se vstupními parametry a proměnnými k pochopení vývojového diagramu.

Tabulka 12 Proměnné funkce visualization()

Název proměnné	Popis
targetAxes	Graf, do kterého se má komponenta vykreslit
segment_struct	Třída komponenty, která je vykreslována (původně pouze struktura)
selectIdx	Index segmentu, který je ve vizualizaci editován (zvýrazněn zeleně)
offset	Offset na ose X, od kterého má být komponenta vykreslována
color	Barva komponenty
currentPos	Výstupní parametr – pozice na ose X, kde skončilo vykreslování

Funkce byla původně vyvinuta pro verifikaci zadaných parametrů v dřívějších fázích projektu. Při ladění modelu bylo potřeba mít jistotu, že jsou zadané rozměry z výkresů správně zapsány. Zadané rozměry v polích se při vyšším počtu segmentů stávaly nepřehlednými. V pozdějších fázích projektu byla funkce upravena a implementována do GUI.

Funkce *assebmlyPreview ()* pak slouží jako skript, který určuje, v jakém pořadí se mají komponenty vykreslit. Volání proběhne pouze při stlačení tlačítka **Show Connections**. Před vykreslením je proveden výpočet pro spojení nábojů s hřídelem. Vypočtené parametry komponent jsou uloženy do dočasných proměnných, které se neukládají do souboru určeného k simulaci. Pokud dojde k chybě během výpočtu z důvodu špatně zadaných parametrů, sestava se nevykreslí.

Ve vykreslené soustavě jsou spojení hřídelů s náboji vykreslena červeně, části s nekonečnou tuhostí zeleně a vnitřní průměry jsou vykresleny růžovou barvou.

6 ZÁVĚR

V rámci práce bylo vyvinuto digitální dvojče osy formového válce flexotiskového stroje pro firmu SOMA. První část práce obsahuje teoretické podklady, ze kterých se při vývoji fyzikálního modelu vycházelo. Byly nalezeny vhodné postupy pro zjednodušení reálného systému do modelu vyvinutého metodou soustředěných parametrů. Pozornost je zaměřena na vývoj validního modelu osy, nikoliv na její vylepšení nebo úpravu konstrukce na základě provedených simulací.

Část práce se věnuje softwaru MATLAB, ve kterém byl model vyvíjen. MATLAB je rozsáhlý systém, proto byla zmíněna pouze témata významná pro vyvíjený model. Kapitola 4 demonstruje základní postupy tvorby modelu v Simulinku s využitím nadstavby Simscape.

V kapitole 5 je detailně popsán vyvinutý model osy formového válce, přičemž jsou použity modelovací postupy z kapitoly 2. Postupy a algoritmy jsou podrobně popsány, aby práce do budoucna mohla posloužit jako výchozí text pro úpravu nebo rozšíření modelu.

Vyvíjený model byl validován na dvou strojích. Model byl vyvíjen na výchozích datech ze stroje typu O20 a data ze stroje typu O14 byla použita pro „slepé testování“- tedy pro simulaci byl do modelu nahrán soubor s parametry pro stroj O14 a byla zkoumána univerzálnost modelu.

U validace měla nejvyšší prioritu první rezonance (tím je míněno nejnižší vlastní frekvence), dále pak první antirezonance, druhá rezonance a třetí rezonance. Výsledky validace z kapitoly 5.12 byly diskutovány s firmou SOMA a byly shledány jako dostatečně přesné a relevantní pro daný účel. Chyba modelu pro typ O14 je ale výrazně menší než pro typ O20. Může to být způsobeno jinou konstrukcí spojky, která musí být při zadávání vstupních parametrů do modelu zjednodušena. Pro typ O14 byla tedy vytvořena jiná aproximace spojky než pro typ O20.

Přesnost modelu by se také dala zvýšit detailnějším modelem motoru. Problémy týkající se nedostatku informací o konstrukci motoru jsou popsány v kapitole 5.6.

Validace rozšíření modelu o bridge byla omezená vhledem k tomu, že není známa přesná torzní tuhost reálného bridge. Byl odhadnut interval, ve kterém by se skutečná hodnota mohla nacházet. Při testech různých torzních tuhostí v tomto intervalu se však chyba modelu výrazně neměnila a výsledky z tabulky 11 lze považovat za validní. Přesné hodnoty parametrů zde nejsou uváděny s ohledem na práva třetích stran. Cílem rozšíření modelu o bridge byla možnost sledování, které parametry mají vliv na dynamiku mechanické soustavy.

Během validace nebylo zohledněno tlumení frekvencí. Pro přesné zachycení skutečného tlumení systému by byla nutná podrobnější analýza materiálových vlastností komponent, o kterých během vývoje modelu nebyl dostatek informací. S parametry tlumení v modelu je ale možné experimentovat a změnou tlumení komponenty lze určit,

jaký má vliv na výslednou frekvenční analýzu. Identifikační metoda navržená v kapitole 5.10 by mohla určit přesnější hodnoty tlumení prvků systému. Z výše uvedených tvrzení lze říct, že celková přesnost modelu by se dala zvýšit s přesnějšími modely komponent od dodavatelů.

Vyvinutý model z kapitoly 5 byl zaveden do užívání ještě před odevzdáním písemné části této diplomové práce, a to pro interní účely a testování hypotéz. Lze tím doložit jeho praktickou použitelnost a užitečnost. Do budoucna bude pravděpodobně upraven vzhled uživatelského prostředí, do kterého je možné přidat funkce, které model nabízí. Celý model byl vyvíjen tak, aby ho bylo možné dále rozšířit o kontakt s dalšími válci a mohl tak být využit pro predikci tiskové chyby. Model je možné implementovat do jiného modelu a simulovat na něm například řízení.

Pro snadnější simulaci a experimenty s modelem bylo vyvinuto uživatelské rozhraní. Z pohledu UX (User experience) byla snaha vyvinou intuitivní a uživatelsky přívětivou aplikaci umožňující snadný návrh parametrů a testování. Z vývojářského hlediska byla aplikace vyvíjena s možností rychlých úprav a jednoduchého rozšíření. Uživatelské rozhraní je v první verzi a může být dále upravováno na základě žádostí uživatelů.

Hlavní výhodou nástroje MATLAB při vypracovávání projektu byla propojenost jeho nástrojů a nadstaveb. Pomocí nástroje App Designer bylo vyvinuto uživatelské prostředí, ze kterého jsou volány funkce napsané ve stejném jazyku, jako je uživatelská aplikace. Vstupní parametry simulace jsou jednoduše získány díky sdílení dat mezi MATLAB pracovními prostředími (workspaces).

7 SEZNAM POUŽITÉ LITERATURY

[1] GRIEVES, Michael. *Origins of the Digital Twin Concept*. 2016/08/31. Dostupné z: doi:10.13140/RG.2.2.26367.61609

[2] MHENNI, Faïda, Ferdinando VITOLO, Andrea REGA, Régis PLATEAUX, Peter HEHENBERGER, Stanislao PATALANO a Jean-Yves CHOLEY. Heterogeneous Models Integration for Safety Critical Mechatronic Systems and Related Digital Twin Definition: Application to a Collaborative Workplace for Aircraft Assembly: Application to a Collaborative Workplace for Aircraft Assembly. *Applied Sciences*. 2022/03/09, **12**, 2787. Dostupné z: doi:10.3390/app12062787

[3] DURÃO, Luiz Fernando C. S., Sebastian HAAG, Reiner ANDERL, Klaus SCHÜTZER a Eduardo ZANCUL. *Digital Twin Requirements in the Context of Industry 4.0*. In: . Cham: Springer International Publishing, 2018, s. 204-214. ISBN 978-3-030-01614-2. Dostupné také z: https://link.springer.com/chapter/10.1007/978-3-030-01614-2_19

[4] WRIGHT, Louise a Stuart DAVIDSON. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*. 2020, **7**(1), 13. ISSN 2213-7467. Dostupné z: doi:10.1186/s40323-020-00147-4

[5] ROSEN, Roland, Georg VON WICHERT, George LO a Kurt D. BETTENHAUSEN. About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine*. 2015, **48**(3), 567-572. ISSN 2405-8963. Dostupné z: doi:https://doi.org/10.1016/j.ifacol.2015.06.141

[6] MATHWORKS. What Is a Digital Twin? *Mathworks* [online]. [cit. 2024-01-08]. Dostupné z: https://www.mathworks.com/discovery/digital-twin.html#digital-twins-with-matlab-and-simulink

[7] MORAVANSKÝ, Lukáš. *Modelování strojů v MapleSim*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství.

[8] CORBO, Mark A. a Stanley B. MELANOSKI. *Practical Design Against Torsional Vibration* [online]. 1996, 189-222 [cit. 2024-02-28]. Dostupné z: https://hdl.handle.net/1969.1/163448

[9] NAI-CHI, LIEN a C YAO GEORGE. IDENTIFICATION OF ANTI-RESONANCE FREQUENCY IN BUILDINGS BASED ON VIBRATION MEASUREMENTS. In: *12th World Conference on Earthquake Engineering* [online]. Auckland, 2000 [cit. 2024-04-16]. Dostupné z: https://www.iitk.ac.in/nicee/wcee/article/1103.pdf

[10] ROBBINS, Michael F. Lumped Element Model. In: CIRCUITLAB, INC. *Ultimateelectronicsbook* [online]. 2021 [cit. 2024-03-03]. Dostupné z: https://ultimateelectronicsbook.com/lumped-element-model/

[11] ANDRÉS, Luis San. *Handout # 1 Modeling of Mechanical (Lumped Parameter) Elements* [Online]. 2008.

[12] MapleSim: Technological Superiority inMulti-Domain Physical Modeling and Simulation. In: *Maplesoft* [online]. 2010 [cit. 2022-02-12]. Dostupné z: https://www.maplesoft.com/contact/webforms/whitepapers/TechSuperiority.aspx

[13] ZHANG, Shengyong. *Vibration Analysis Projects of Lumped-Parameter and Distributed-Parameter Systems*. Minneapolis, MN: ASEE Conferences, 2022/08/23. Dostupné z: doi:10.18260/1-2--40491

[14] NAVARRO, Lucas a Luiz GOES. *Aircraft Braking Dynamics and Brake System Modeling for Fault Detection and Isolation*. 2019/01/01. Dostupné z: doi:10.26678/ABCM.DINAME2019.DIN2019-0181

[15] MATHWORKS. Flexible Shaft. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/sdl/ref/flexibleshaft.html

[16] NESTORIDES, E.J. *A Handbook On Torsional Vibration* [online]. Cambridge., At The University Press, 1958 [cit. 2024-03-04]. Dostupné z: https://archive.org/details/in.ernet.dli.2015.140286/mode/2up

[17] ZUPANCIC, Borut, Rihard KARBA, Maja ATANASIJEVIC-KUNC a Josip MUSIĆ. Continuous Systems Modelling Education – Causal or Acausal Approach? In: *Proceedings of the ITI 2008 30th International Conference on Information Technology Interfaces*. IEEE Computer Society, 2008, s. 803-808. ISBN 978-953-7138-12-7.

[18] MATHWORKS. Double Mass-Spring-Damper in Simulink and Simscape. *Mathworks* [online]. c1994-2024 [cit. 2024-05-15]. Dostupné z: https://www.mathworks.com/help/simscape/ug/double-mass-spring-damper-in-simulink-and-simscape.html

[19] KLEIJNEN, Jack P.C. Verification and validation of simulation models. *European Journal of Operational Research*. 1995, **82**(1), 145-162. ISSN 0377-2217. Dostupné z: doi:https://doi.org/10.1016/0377-2217(94)00016-6

[20] DAVIS, Paul. *Generalizing Concepts and Methods of Verification, Validation, and Accreditation (VV&A) for Military Simulations*. 1992/01/01, 57.

[21] HAIGH, Thomas. Cleve Moler: Mathematical Software Pioneer and Creator of Matlab. *IEEE Annals of the History of Computing* [online]. 2008, **30**(1), 87-91 [cit. 2024-01-06]. ISSN 1058-6180. Dostupné z: doi:10.1109/MAHC.2008.2

[22] MATHWORKS. Base and Function Workspaces. *Mathworks* [online]. c1994-2024 [cit. 2024-05-08]. Dostupné z:

https://www.mathworks.com/help/matlab/matlab_prog/base-and-function-workspaces.html

[23] MATHWORKS. Model Workspaces. *Mathworks* [online]. c1994-2024 [cit. 2024-05-08]. Dostupné z: https://www.mathworks.com/help/simulink/ug/using-model-workspaces.html

[24] MATHWORKS. Assignin. *Mathworks* [online]. c1994-2024 [cit. 2024-05-08]. Dostupné z: https://www.mathworks.com/help/matlab/ref/assignin.html

[25] MATHWORKS. Compare Solvers. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/simulink/ug/compare-solvers.html

[26] MATHWORKS. Variable Step Solvers in Simulink. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/simulink/ug/variable-step-solvers-in-simulink-1.html

[27] MATHWORKS. Choose a Solver. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/simulink/ug/choose-a-solver.html

[28] MATHWORKS. Rotational Electromechanical Converter. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/simscape/ref/rotationalelectromechanicalconverter.html

[29] MATHWORKS. Simscape Multibody. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/sm/

[30] MATHWORKS. Simscape Driveline. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/sdl/

[31] MATHWORKS. Ideal Rotational Motion Sensor. *Mathworks* [online]. c1994-2024 [cit. 2024-05-09]. Dostupné z: https://www.mathworks.com/help/simscape/ref/idealrotationalmotionsensor.html

[32] MATHWORKS. Variant Connector. *Mathworks* [online]. c1994-2024 [cit. 2024-05-20]. Dostupné z: https://www.mathworks.com/help/simscape/ref/variantconnector.html

[33] MATHWORKS. Mass-Spring-Damper with Controller. *Mathworks* [online]. c1994-2024 [cit. 2024-05-01]. Dostupné z: https://www.mathworks.com/help/simscape/ug/mass-spring-damper-with-controller.html

[34] MATHWORKS. Ideal Translational Motion Sensor. *Mathworks* [online]. c1994-2024 [cit. 2024-05-01]. Dostupné z: https://www.mathworks.com/help/simscape/ref/idealtranslationalmotionsensor.html

[35] MATHWORKS. Ideal Force Source. *Mathworks* [online]. c1994-2024 [cit. 2024-05-01]. Dostupné z: https://www.mathworks.com/help/simscape/ref/idealforcesource.html

[36] BOSCH REXROTH AG. *MS2N Synchronous servo motors* [Online]. 2017.

[37] ZHANG, D., W. GAO a X. YAN. Determination of Natural Frequencies of Pipes Using White Noise for Magnetostrictive Longitudinal Guided-Wave Nondestructive Testing. *IEEE Transactions on Instrumentation and Measurement*. 2020, **69**(6), 2678-2685. ISSN 1557-9662. Dostupné z: doi:10.1109/TIM.2019.2931528

[38] MATHWORKS. Band-Limited White Noise. *Mathworks* [online]. c1994-2024 [cit. 2024-05-22]. Dostupné z: https://www.mathworks.com/help/simulink/slref/bandlimitedwhitenoise.html

[39] MATOUSEK, Radomil, S LANG, Petr MINAR a Petr PIVOŇKA. Evolutionary Design of Polynomial Controller. *World Academy of Science, Engineering and Technology, International*. 2011, **5**, 2407-2412. Dostupné také z: https://api.semanticscholar.org/CorpusID:42057518

8 SEZNAM OBRÁZKŮ

9 SEZNAM TABULEK

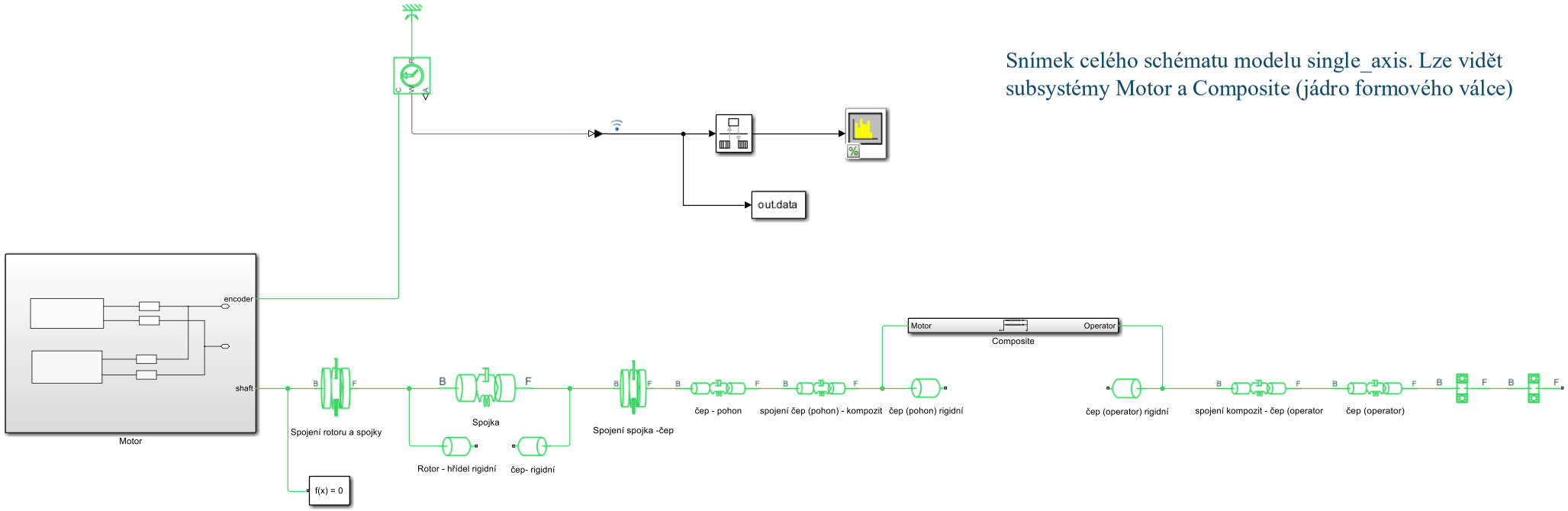
10 SEZNAM PŘÍLOH

Příloha I. - Snímek celého blokového schématu
Příloha II. - Snímek modelovaného motoru blokovým schématem
Příloha III. - Snímek blokového schématu segmentace bridge
Příloha IV. - Snímek modelovaného nasazení bridge na jádro

Elektronické přílohy
Zip soubor obsahující:

- Soubory potřebné pro simulaci modelu (třídy, funkce...)
- Soubor se vstupními daty (fiktivní parametry)
- Instalační soubor vytvořené aplikace do MATLABu
- Snímky modelu

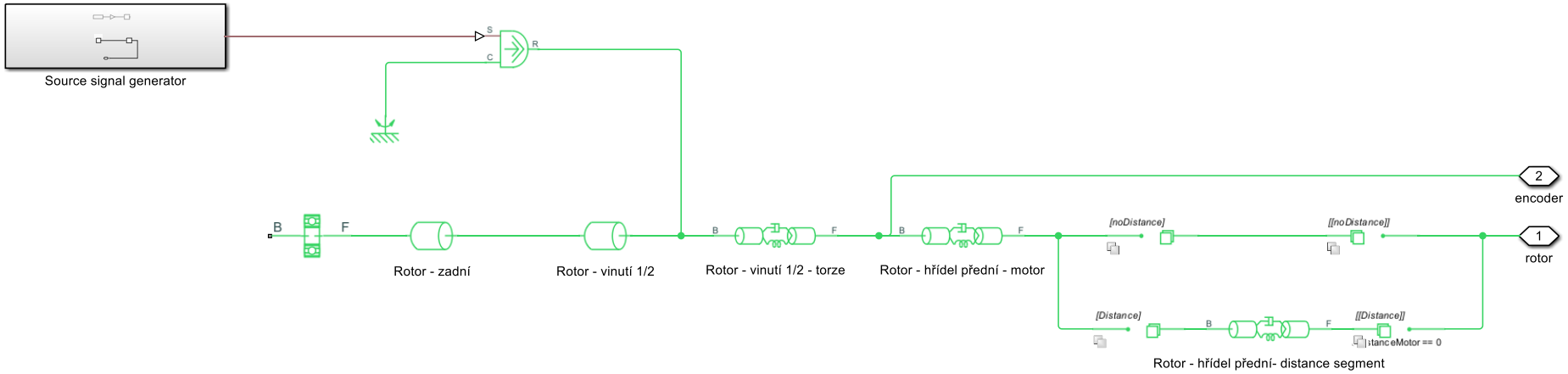
PŘÍLOHA I.



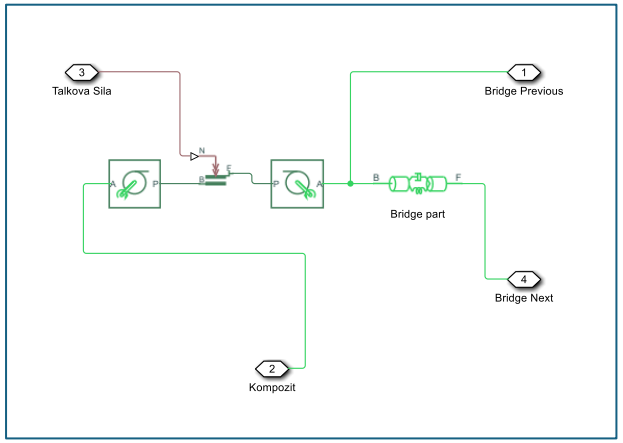
Snímek celého schématu modelu single_axis. Lze vidět subsystémy Motor a Composite (jádro formového válce)

PŘÍLOHA II.

Snímek celého schématu subsystému *Disk model of rotor*, který je součástí subsystému *Motor*. Zapojené schéma se vztahuje k obrázku č. 28.

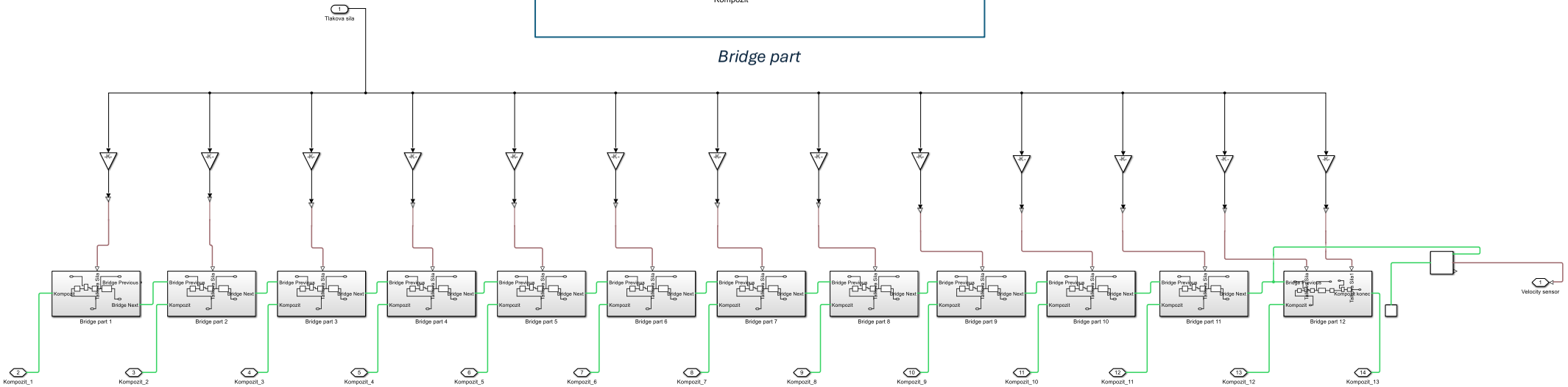


PŘÍLOHA III.



Bridge part

Snímek schématu subsystému s rozdělením bridge do segmentů s detailem na subsystém *Bridge part*.



PŘÍLOHA IV.

Snímek schématu subsystému *kompozit - bridge*, který je součástí subsystému *Composite*. Lze vidět zapojení bloku *Scope* pro verifikaci (kapitola 5.11).