

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020

Bc. Michal Husák



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

VIRTUÁLNÍ DVOJČE PRO TESTBED PRŮMYSLU 4.0

VIRTUAL TWIN FOR TESTBED INDUSTRY 4.0

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Husák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2020

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Michal Husák

ID: 188985

Ročník: 2

Akademický rok: 2019/20

NÁZEV TÉMATU:

Virtuální dvojče pro testbed Průmyslu 4.0

POKYNY PRO VYPRACOVÁNÍ:

Vyberte vhodný nástroj pro realizaci virtuálního obrazu testbedu průmyslu 4.0.

- 1) Zpracujte rešerši popisující vhodnost využití softwarových nástrojů Tecnomatix Process Simulate a Mechatronics Concept Designer na vytvoření digitálního obrazu testbedu.
- 2) Na demonstrátoru Průmyslu 4.0 vytipujte vhodnou výrobní či transportní jednotku a proveďte její propojení s vybraným systémem. Ověřte toto řešení.
- 3) Vytvořte metodiku popisující integraci dalších buněk testbedu do zvoleného simulačního nástroje, případně tuto integraci proveďte.
- 4) Vyhodnoťte navržené a realizované řešení.

DOPORUČENÁ LITERATURA:

Uživatelská příručka Tecnomatix Process Simulate.

Dokumentace k platformě Mechatronics Concept Designer

Termín zadání: 3.2.2020

Termín odevzdání: 1.6.2020

Vedoucí práce: Ing. Václav Kaczmarczyk, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato diplomová práce se zabývá tvorbou digitálního dvojčete pro testbed Barman. Barman je školní model autonomní linky míchaných nápojů, který demonstruje principy Průmyslu 4.0. Teoretická část práce diskutuje nad volbou vhodného nástroje pro virtualizaci. Porovnává nástroje Tecnomatix Process Simulate a modul Mechatronic Concept Designer integrovaný v prostředí NX. Praktická práce je dělena do dvou fází. V první fázi práce byl hledán způsob integrace robotického SCARA manipulátoru. Druhá fáze se zaměřuje na virtualizaci buňky Shaker pomocí druhého ze zmíněných nástrojů. Tato práce je koncipována jako návod tvorby a ověření konceptu digitálního dvojčete.

Klíčová slova

Průmysl 4.0, Virtuální zprovoznění, Digitální dvojče, Tecnomatix Process Simulate, Mechatronics Concept Designer, Shaker, Robot, Barman, PLCSIM Advanced

Abstract

The goal of my master's thesis is to create a digital twin of a testbed Barman. The Barman is a school model of autonomous mixed drinks production line that demonstrates the principles of Industry 4.0. In the theoretical part of the thesis, the choice of a suitable tool for virtualization is discussed. The Tecnomatix Process Simulate and the Mechatronic Concept Designer module integrated in the NX platform is compared. The practical part of the work is divided into two phases. The first phase was about looking for a way to integrate the robotic SCARA manipulator. The second phase was focused on the virtualization of the cell Shaker in the latter of the tools mentioned before. This work is designed as a guide for creating and verifying the concept of a digital twin.

Keywords

Industry 4.0, Virtual commissioning, Digital twin, Tecnomatix Process Simulate, Mechatronics Concept Designer, Shaker, Robot, Barman, PLCSIM Advanced

Bibliografická citace:

HUSÁK, Michal. *Virtuální dvojče pro testbed Průmyslu 4.0*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/126919>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Václav Kaczmarczyk.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Virtuální dvojče pro testbed Průmyslu 4.0 jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2020**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Václavu Kaczmarczykovi, Ph.D. a celé Skupině průmyslové automatizace za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **20. května 2020**

.....
podpis autora

Obsah

1.	Úvod.....	14
2.	Koncepty Průmyslu 4.0.....	15
2.1.1	Internet věcí	16
2.1.2	Aditivní výroba	16
2.1.3	Přínosy a důsledky I4.0.....	16
2.2	Virtuální zprovoznění.....	17
2.2.1	Koncepce	18
2.2.1.1	Hardware in the Loop.....	18
2.2.1.2	Software in the Loop	19
2.2.2	Průzkum trhu.....	19
2.3	Tecnomatix Process Simulate	21
2.3.1	Možnosti externího řízení TPS	21
2.4	Mechatronics Concept Designer	22
2.4.1	Možnosti externího řízení MCD	23
2.5	Porovnání MCD a TPS.....	24
2.5.1	Propojení.....	24
2.5.2	Flexibilita.....	24
2.5.3	Robotické simulace.....	25
2.6	SIMIT.....	25
2.7	OPC UA	26
2.8	TestBed Barman.....	27
2.8.1	Dopravník	28
2.8.2	Manipulátor.....	28
2.8.3	Sklad sklenic	28
2.8.4	Drtič ledu	29
2.8.5	Zásobník nealkoholických nápojů	29
2.8.6	Zásobník alkoholických nápojů	29
2.8.7	Šejkr.....	30
2.8.8	Sodobar	30
3.	Virtualizace	31

3.1	Práce v TPS	31
3.1.1	Modely	31
3.1.1.1	Stůl.....	32
3.1.1.2	Buňky	34
3.1.2	Robot.....	35
3.1.3	Efektor	39
3.1.4	Konfigurace externího připojení.....	40
3.2	Práce v MCD.....	41
3.2.1	Struktura projektu	41
3.2.1.1	Vlastníci.....	42
3.2.2	Nedeformovatelná tělesa.....	43
3.2.3	Kolidující tělesa	44
3.2.4	Materiály simulace.....	45
3.2.4.1	Hmotnost tištěných dílů.....	45
3.2.5	Spoje a vazby	47
3.2.6	Senzory a aktuátory	48
3.2.7	Signály a mapování.....	52
3.2.8	Externí signály	54
3.2.9	Mapování signálů.....	56
3.3	Řízení virtuálního modelu.....	57
3.3.1	Řídicí program SiL	57
3.3.2	Řídicí program HiL.....	58
3.3.3	Virtualizace PLC.....	59
3.4	Ověření funkce	60
3.5	Řešené problémy	61
3.5.1	Problémy s TPS	61
3.5.2	Problémy s MCD	61
3.6	Inverzní kinematika.....	63
4.	Závěr	64

Seznam symbolů a zkratk

Zkratky:

I4.0	...	Industry 4.0
IoT	...	Internet of Things
š/h/v	...	šířka/hloubka/výška
RFID	...	Radio Frequency Identification
FDM	...	Fused Deposition Modeling
FFF	...	Fused Filament Fabrication
PLM	...	Product Lifecycle Management
VD	...	Virtuální dvojče
HiL	...	Hardware in the Loop
SiL	...	Software in the Loop
PLC	...	Programmable Logic Controller
HMI	...	Human Machine Interface
TIA	...	Totally Integrated Automation
CAD	...	Computer Aided Design
CAM	...	Computer Aided Manufacturing
CAE	...	Computer Aided Engineering
CNC	...	Computer Numerical Control
TPS	...	Tecnomatix Process Simulate
MCD	...	Mechatronics Concept Designer
CEE	...	Cyclic Event Evaluation
OPC	...	Open Process Control
UA	...	Unified Architecture
DA	...	Data Access
SHM	...	Shared Memory Connection
M2M	...	Machine To Machine
TCP	...	Transmission Control Protocol
IP	...	Internet Protocol
UDP	...	User Datagram Protocol
HTTP	...	Hypertext Transfer Protocol

SOAP	...	Simple Object Access Protocol
JT	...	Jupiter Tessellation
COJT	...	Component Jupiter Tessellation
STEP	...	Standard for the Exchange of Product Data
OLP	...	Open Location Platform
C#	...	C Sharp
SCARA	...	Selective Compliance Assembly Robot Arm
CSYS	...	Coordinate System
XLSX	...	formát souboru Office Open XML
XML	...	Extensible Markup Language
SI	...	International System of Units
PWM	...	Pulse Width Modulation
DB	...	Data Block
FB	...	Function Block
IO	...	Input/Output

Seznam obrázků

Obr. 1 Průmysl 4.0 [3]	15
Obr. 2 Hardware in the Loop [4][5].....	18
Obr. 3 Software in the Loop [4].....	19
Obr. 4 Možnosti propojení MCD [6]	23
Obr. 5 TestBed Barman	27
Obr. 6 Buňka sklad sklenic vlevo, buňka drtič ledu vpravo	28
Obr. 7 Buňka sklad alkoholických nápojů.....	29
Obr. 8 Buňka šejkr vlevo, buňka sodobar vpravo.....	30
Obr. 9 Funkce zjednodušení sestavy v NX.....	32
Obr. 10 Nastavení importu modelu.....	33
Obr. 11 Souřadnice buněk na stole v TPS	33
Obr. 12 V levé části model malé buňky, v pravé části model velké buňky.....	34
Obr. 13 Model robota EPSON	35
Obr. 14 Signály robota.....	36
Obr. 15 Editace bodů operace	36
Obr. 16 Editace robotického programu.....	37
Obr. 17 Náhled na body a operace robota	37
Obr. 18 Vytvoření signálů efektoru	39
Obr. 19 Nastavení externího spojení.....	40
Obr. 20 Signály potřebné k simulaci	40
Obr. 21 Stromová struktura projektu	41
Obr. 22 Organizace vlastníků	42
Obr. 23 Nedeformovatelné těleso	43
Obr. 24 Kolidující těleso.....	44
Obr. 25 Určení hmotnosti 3D tištěných těles.....	46
Obr. 26 Mechanické spoje a vazby	47
Obr. 27 Senzor kolize (bezkontaktní snímač polohy).....	48
Obr. 28 Limitní senzory	49

Obr. 29 Řízení pozice solenoidu	50
Obr. 30 Komponenta vlastníka a změna vzhledu	51
Obr. 31 Náhled adaptéru signálu	53
Obr. 32 Externí konfigurace signálu	55
Obr. 33 Mapování signálů	56
Obr. 34 Koncept HiL	58
Obr. 35 PLCSIM Advanced.....	59
Obr. 36 Schéma konceptu	60
Obr. 37 Inverzní kinematika v MCD	63

Seznam tabulek

Tab. 1 Průzkum trhu	20
Tab. 2 Produktová základna firmy Siemens [5]	20
Tab. 3 Souřadnice buněk v TPS.....	34
Tab. 4 Koncové body operací	38
Tab. 5 Robotické programy v TPS	38
Tab. 6 Signály buňky Shaker	52
Tab. 7 Možnosti propojení s PLC	57
Tab. 8 Nedeformovatelná tělesa	69

1. ÚVOD

Tématem této práce je průzkum dostupných možností realizace digitálního dvojčete stroje projektu TestBed Barman. Tento projekt je demonstrátorem konceptu Průmysl 4.0, jehož neodmyslitelnou součástí je i jeho digitální realizace. Cílem je proto porovnat dostupné programové nástroje a nalézt vhodnou cestu integrace digitálních obrazů výrobních a transportních buněk stroje. Projekt Barman si klade za cíl implementovat stávající známé postupy nazývané obecně čtvrtou průmyslovou revolucí a dále je rozvíjet.

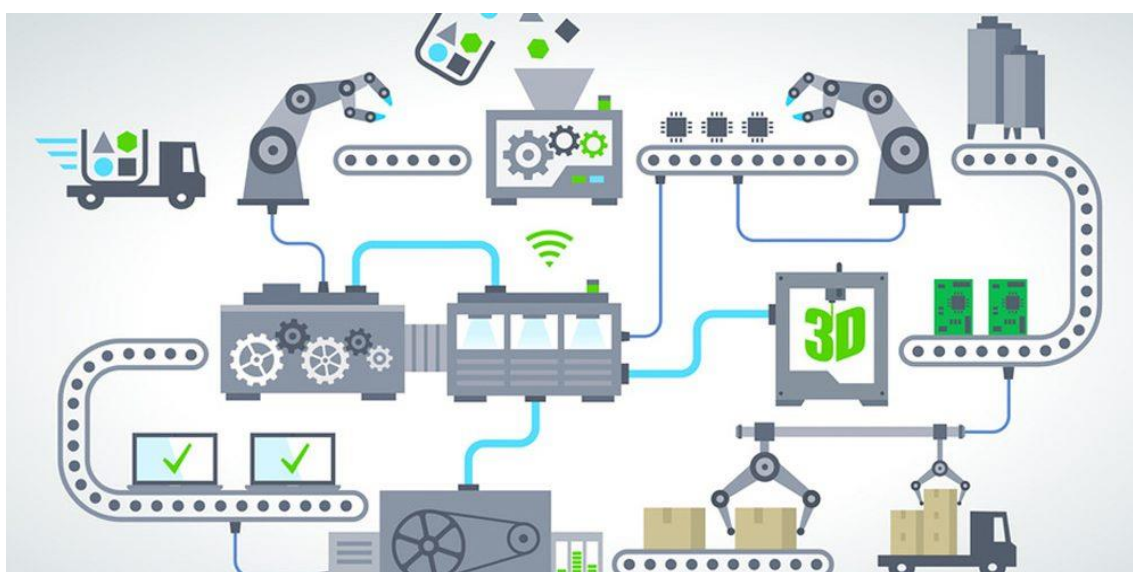
Všeobecně rozšířený pojem Průmysl 4.0, je horké téma poslední dekády. Vysoké nároky na výrobu jsou důsledkem rozšiřujícího se trhu, který stále více klade důraz na variabilitu a rychlost výroby. Továrny chrlící statisícové série stále více ztrácí svůj potenciál a přednost dostávají malé série „šité na míru“ zákazníka. Koncept čtvrté průmyslové revoluce propojuje výhody horizontální integrace výrobních kapacit s vertikální integrací zdrojů a zákazníků. Veškerá data jsou strukturována v cloudových službách dostupné pro zařízení na jakékoliv úrovni. Jednou z částí tohoto konceptu je i digitální dvojče.

Zadání této práce vychází z požadavku zatím otevřené koncepce virtuálního zprovoznění projektu Barman. Z dostupných řešení se nabízely dva nástroje. Jedním nástrojem byl Tecnomatix Process Simulate a druhým Mechatronic Concept Designer. Oba pochází z dílny firmy Siemens, která drží vůdčí pozice na poli virtualizace a Průmyslu 4.0. Zásadní otázkou bylo, jakého konceptu virtuálního zprovoznění jsme schopni dosáhnout s použitým hardwarem a softwarem. Druhou otázkou bylo, k čemu jsme aplikovaná řešení schopni uplatnit.

V práci má být vytypována transportní jednotka nebo výrobní buňka, která bude virtuálně realizována. Výsledek má sloužit jako návod, který lze aplikovat na zbylé části stroje. Z toho důvodu si práce klade za cíl, být co nejvíce popisná a vyobrazit možnosti i úskalí výše zmíněných nástrojů. Hlavní důraz je kladen na finální část v MCD, která má i praktické výstupy pro projekt Barman.

2. KONCEPTY PRŮMYSLU 4.0

Průmysl 4.0 nebo anglicky Industry 4.0, zkráceně I4.0, je pojmenování nové koncepce průmyslové výroby, která je často označována jako čtvrtá průmyslová revoluce. První průmyslovou revolucí byl vynález a zavedení parního stroje do výroby. Za druhou revolucí je označována pásová výroba, masová elektrifikace a mobilita. Třetí revolucí je příchod počítačů do průmyslové výroby. Průmysl 4.0 je revolucí v přístupu k průmyslové výrobě, jejímu řízení. Zabývá se úplnou automatizací výroby bez potřeby zásahu člověka a jejím návrhem. Je založena na vzájemně interagujících autonomně řízených kyberneticko-fyzických systémech. [1]



Obr. 1 Průmysl 4.0 [3]

Koncept I4.0 byl poprvé představen veřejnosti na hanoverském veletrhu v roce 2011. Na této konferenci byl nastíněn směr a čtyři základní pilíře: kyber-fyzické systémy, internet věcí, internet služeb a digitální ekonomika. Přínosem této revoluce by mělo být odstranění duševní práce ve výrobě a její organizace. [1]

Průmysl 4.0 přichází s definicí prostředí takzvané „chytré továrny“ s plnou autonomií na člověku. S ohledem na dnešní požadavky zákazníků, na maximální flexibilitu výroby a personalizaci výrobků je v krajním případě možné, aby z výrobního pásu nesjely dva totožné výrobky. Vzniká proto potřeba, aby v chytré továrně výrobky samy rozhodovaly o svém výrobním procesu a tím definovaly svůj výsledný vzhled, vlastnosti nebo i parametry. Toho lze docílit díky digitalizaci výroby. Výrobek musí být přesně popsán, například virtuálním obrazem. Další požadovanou vlastností je distribuce řízení výrobních procesů do autonomních, takzvaných kyberneticko-fyzikálních systémů. To znamená, že každá část výrobní linky je schopna rozhodovat sama za sebe a koordinovat svoji činnost s ostatními. Aby byla taková kooperace možná, musí být tyto systémy propojeny. Základem sdílení dat je meziúrovňové propojení a principy IoT (*Internet of Things*). [1]

Hlavní požadavky průmyslu 4.0 na zařízení jsou:

- vzájemné propojení a komunikace (internet věcí)
- virtuální obraz fyzických zařízení a možnost jejich nezávislé simulace i vzájemné porovnání výsledků
- schopnost zařízení rozhodovat o tom co bude vyrábět
- rychlost rozhodování
- modularita, schopnost pružně reagovat na měnící se požadavky

2.1.1 Internet věcí

Často označován zkráceně IoT (*Internet of Things*) je jeden z konceptů integrovaných do I4.0. Za IoT zařízení považujeme takové, které dokáže sdílet svoje provozní stavy a parametry na síť. Poskytuje možnost jiným zařízením nebo nadřazeným systémům sledovat nebo ovlivňovat jejich chod. Sdílené informace jsou dostupné pro všechny úrovně řídicích systémů díky meziúrovňovému propojení. Toho využívají kyber-fyzické systémy k decentralizovanému rozhodování v chytré továrně. Tyto systémy mají svůj virtuální obraz. Takový obraz nemusí být jen 3D model, ale je to například soubor informací o výrobku na RFID čipu. [1]

2.1.2 Aditivní výroba

V průmyslu 4.0 je také hojně využíváno technologie aditivní výroby na 3D tiskárnách. Vlastnosti 3D tiskáren se významně shodují s požadavky chytré továrny. S masovým rozšířením a zlevněním 3D tisku po vypršení patentů je vyřešena potřeba snadného a rychlého prototypování strojů, části strojů nebo výrobků. Můžeme tak docílit agilnější realizace projektu nebo přímo začlenit 3D tisk do výroby. Kupříkladu stroj, kterým je tato práce podepřena, je z velké části vytvořen na 3D tiskárně technologií FDM (Fuse Depositing Modeling). [1]

Díky tomu je při vývoji stroje možné snadno aplikovat metody Rapid Prototyping. V překladu výraz znamená rapidní, nebo také rychlé prototypování. Jedná se o metodický postup při návrhu nového produktu. Skládá se ze čtyř kroků: analýza, design, verifikace a výroba. Kroky jsou zacykleny a jednotlivé cykly se nazývají iterace. Výroba je tedy reprezentována 3D tiskem. [1]

2.1.3 Přínosy a důsledky I4.0

Celková integrace zde popsaných principů se nazývá chytrou továrnou. Takové zařízení by mělo být schopno naplno využít svého výrobního potenciálu za minimálního fyzického působení člověka. V dnešních dnech, působení koronavirové pandemie, jsou výhody čtvrté průmyslové revoluce patrné. Stavět „hloupé“ továrny už nebude žádoucí, protože nejsou schopny pružně reagovat na aktuální požadavky trhu. [2]

Příchodem čtvrté průmyslové revoluce vyvstávají otázky a rizika, které budou nutné ještě vyřešit. Jsou to otázky sociální, ekonomické a také bezpečnostní. Společnost se bude muset připravit na velké změny na pracovním trhu, zabezpečení lidí proti zranění strojem, tak i stojí proti poškození lidmi. V neposlední řadě se počítá s velkými ekonomickými důsledky. [2]

2.2 Virtuální zprovoznění

Jedním z důležitých principů průmyslu 4.0 je virtuální zprovoznění. Jejím cílem je vytvořit virtuální obraz továrny, která realizuje digitální výrobu. Virtuální továrnou se myslí nefyzický obraz výrobních prostředků, které reprezentují dynamiku, výrobní kapacitu i dobu života fyzického zařízení. Takové modely nazýváme virtuální dvojčata, dále jen VD. Slouží k ověření koncepce stroje při jeho vývoji, optimalizaci výrobních procesů a výuky obsluhy. Digitální výroba na VD reprezentuje celý výrobní postup výrobku. Popisuje vstupní materiál, jednotlivé výrobní kroky i dobu života výrobku. Vlastnosti a parametry získané virtuálními metodami lze využít k vylepšení, identifikaci nebo plánování životního cyklu produktu (PLM). Budoucností v tomto oboru je integrace moderních metod umělé inteligence a strojového učení pro predikci poruch. Plánovaná údržba má zamezit neplánovaným odstávkám výrobních zařízení a prodloužit životní cyklus stroje. [1]

V různých průmyslových odvětvích se VD používají k optimalizaci provozu, údržbě hardwaru řídicích systémů a výrobních procesů. VD je také součástí kyber-fyzických systémů, které je využívají k plánování. Důležitým faktorem VD je takzvané „přesimulování“. Jedná se o stav, kdy virtuální model kopíruje realitu, to je jednak složité na výpočetní kapacity modelu a za druhé generuje velké množství dat. U virtuálního dvojčete není potřebná přesná emulace, nýbrž je možné model zjednodušit. Úkolem integrátorů takových systémů je tak zvolit optimální zjednodušení. [1]

Klíčové výhody virtuálního zprovoznění stroje jsou:

- Ověření konceptu stroje a jeho řídicích algoritmů
- Úspora času inženýra
- Ověření senzorů a jejich logiky
- Testy blokad a bezpečnostních prvků
- Ošetření chybových stavů
- Snadná off-line optimalizace
- Snazší ožívání stroje

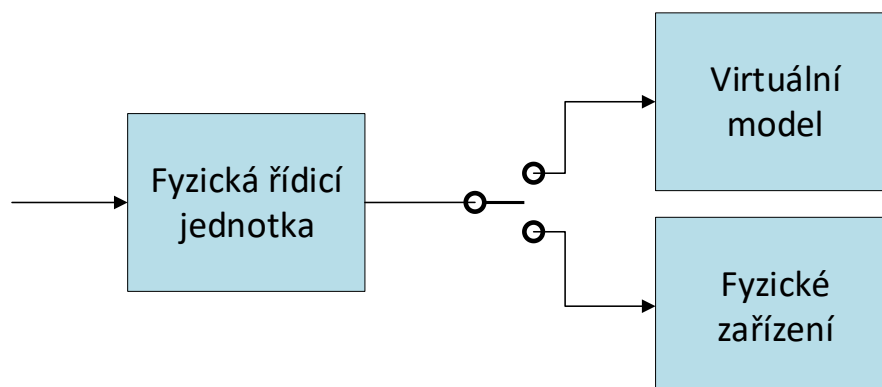
2.2.1 Konceptce

Nejprve je nutné ujasnit si pojmy simulace a emulace. Simulace je snaha zachytit zjednodušenou podstatu funkce zařízení nebo situace. Oproti tomu emulace kopíruje přesnou funkci. Podoba simulace nebo emulace může být fyzická nebo virtuální. Z těchto pojmů vychází i název virtualizace neboli virtuální simulace, který je v této práci často používán. Opakem tomu je fyzická realizace, například simulační model. [1]

Realizace virtuálního nebo fyzického zprovoznění spolu koncepčně souvisí. Jsme schopni definovat oba řídicí řetězce a jejich části kombinovat v různé koncepty. Pro virtuální zprovoznění jsou zásadní dvě kombinace propojení. První je kombinace fyzického řídicího systému a virtuálního modelu. Tato koncepce se jmenuje „*Hardware in the Loop*“. Druhou koncepcí je takzvaná metoda „*Software in the Loop*“, která propojuje virtuální model i s virtuálním kontrolérem. Nevirtuální metody používají úplný fyzický hardware nebo pouze virtuální kontrolér. [4]

2.2.1.1 Hardware in the Loop

Hloubka virtualizace záleží na požadavcích funkce virtuálního dvojčete. Nejnižší stupeň je virtuální obraz zařízení. Pokud takový obraz nebo model řídíme reálným PLC, hovoříme o virtuálním zprovoznění koncepce „*Hardware in the Loop*“ (*HiL*). Tato metoda virtualizace je nejdůležitější při tvorbě řídicího programu a ověření koncepce stroje. Výhodou je, že fyzické zařízení ještě nemusí existovat a týmy programátorů mohou tvořit řídicí rutiny. Nehrozí nebezpečí zničení drahého hardwaru. Řídicí program je dále optimalizován. Po vyrobení fyzického stroje postačí v ideálním případě pouze nahrát příslušné PLC, nebo HMI a stroj je připraven k funkčním testům. I po uvedení fyzického zařízení do provozu má tato koncepce své uplatnění. Jak je vidět na obrázku Obr. 2, virtuální model zde slouží pro ověřování nových optimalizačních úloh. Dále je model vhodný pro výcvik údržby a dispečerů. U nestabilních systémů je takovýto model jedinou možností bezpečného ověření nových algoritmů nebo obsluhy. [4]

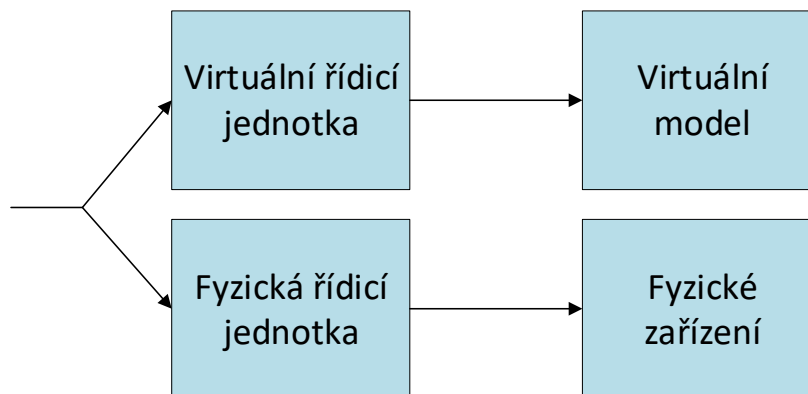


Obr. 2 Hardware in the Loop [4][5]

2.2.1.2 Software in the Loop

Dalším stupněm virtualizace je simulace nebo emulace samotného řídicího PLC. Takovou konfiguraci označujeme za virtuální výrobu koncepce „*Software in the Loop*” (*SiL*). Není potřeba řešit přepínání jako u *SiL*. Virtuální část běží s fyzickou částí paralelně. Při výrobním procesu získáme jak fyzický výrobek, tak i jeho virtuální kopii. Virtuální klon výrobku slouží jako výrobní dokumentace reálného nebo jako vstup do dalších systémů kontroly kvality. Systém je vhodný pro úplné off-line plánování výroby. [4]

Systémy pro řízení a plánování výroby mohou na tomto modelu ověřovat své výrobní postupy, optimalizovat výrobní čas a využití kapacity výroby. Další významnou výhodou systému, který má vlastní virtuální obraz, je i možnost porovnávání výsledků virtuální a fyzické výroby. [4]



Obr. 3 Software in the Loop [4]

2.2.2 Průzkum trhu

Na trhu je možné vybrat si z více softvérových nástrojů pro virtuální zprovoznění. Společnosti známých jmen se snaží integrovat tyto koncepce do svých portfolií. Do tohoto úkolu se většinou pouští firmy, které se zaměřují na CAD platformy. Konstrukční programy jsou základním kamenem virtualizace. Pokud je tedy virtualizační nebo simulační balíček implementován, odpadá nutnost konverze modelů. Příkladem jsou firmy Dassault Systemes nebo Autodesk se svými produkty.

Významnou roli hrají i výrobci automatizační techniky. Ke svým produktům poskytují i softwarovou podporu pro tvorbu virtualizace. Je otázkou, jak otevřené jsou jejich platformy i k použití konkurenčního hardwaru. Mezi zástupce tohoto odvětví patří například firmy Beckhoff, ABB nebo Siemens.

Dalšími hráči na poli virtuálního zprovoznění jsou v neposlední řadě i společnosti zabývající se herním průmyslem. Příkladem může být herní studio Unity se svým programem Perspective.[5]

Tab. 1 Průzkum trhu

Značka	Software pro virtualizaci	Vlastní Hardware	CAD soft. podpora
Beckhoff	TwinCAT	ano	ne
ABB	RoboStudio	ano	ne
Siemens	NX MCD, Process/Plant Simulate	ano	NX
Rockwell Automation	Emulate3d	ano	ne
Dassault Systemes	Delmia	ne	SolidWorks
Autodesk	Forge	ne	Catia, Inventor
Visual Components	Visual Components	ne	ne
Unity	Perspective	ne	ne

Například software SolidWorks disponuje modulem pro snadný export modelů do Emulate3d. Úplnou integraci provedla firma Siemens ve svém produktu NX. I to je jeden z důvodů, proč je finální část této práce věnována právě tomuto softwaru.

Tab. 2 Produktová základna firmy Siemens [5]

Úroveň integrace	továrna	linka	stroj
Ověřované aspekty	Produktivita, propustnost	Řídicí program (PLC, robot) čas cyklu, kolize	Posloupnost operací, řídicí program (PLC, NC), funkce, fyzikální jevy
Časové měřítko simulace	dny	hodiny	minuty
Úroveň abstrakce	Diskrétní události	Simulace založená na čase a událostech, 3D kinematika, detekce kolizí	Simulace založená na čase a událostech, 3D pohyb, kolize
Řešení Siemens	Plant Simulation	Process Simulation	Mechontronic Concept Designer

Výhodou firmy Siemens je, že disponují vlastním CAD platformou NX. Modul MCD je do ní plně integrován a tím odpadá potřeba konverze modelů. Zrychluje se tím proces prototypování. Dále je optimalizované propojení s ostatními simulačními nástroji stejné značky například SIMIT nebo PLCSIM Advanced. Portfolio značky rozděluje aplikace softwaru podle časové náročnosti. Pro delší a dlouhé časové simulace z pohledu výroby má Siemens k dispozici softwarový balík Tecnomatix. [5]

2.3 Tecnomatix Process Simulate

Tecnomatix je balík komplexních softwarů realizujících digitální řešení výroby od firmy Siemens PLM Software. Zjednodušuje synchronizaci mezi výrobkem, výrobním inženýrstvím, výrobou a servisem. Díky tomu je možné maximalizovat efektivitu výroby a snižovat náklady na výrobu. Řešení Tecnomatix je založeno na otevřené architektuře a tím dává možnost aplikace prakticky u jakéhokoliv systému. Z balíku je v této práci využíván jeden software vhodný pro simulaci výroby, a to je Process Simulate. [5][9]

Tecnomatix Process Simulate dále jen TPS, je jeden z balíku simulačních softwarů, který poskytuje nástroj pro digitalizaci výrobních strojů a linek. Poskytuje 3D prostředí pro vizualizaci a možnou kontrolu procesů, dosahu i bezpečné vzdálenosti stroje. Umožňuje kontrolu nad 3D daty produktů a zdrojů, aby bylo možné validovat a optimalizovat složité výrobní procesy. Můžeme si zde ověřit většinu výrobních procesů jako např. montáž, svařování, kontinuální procesy, jako je laserové svařování, lepení, lakování a další robotické procesy. Navíc je v TPS možné simulovat přítomnost člověka a jeho ergonomii. Díky tomu je možné vytvářet bezpečné a také pro páci pohodlné pracovní podmínky. [10]

Klíčovou vlastností TPS je práce s roboty. Software poskytuje významnou podporu pro tvorbu robotických linek. [5]

- Optimální umístění robota do pracovní plochy
- Výpočet kolizí i dosažitelnosti
- Kolaborace robotů
- Off-line programování robotů
- Teach-in metody
- Upload/Download programu z a do robota

2.3.1 Možnosti externího řízení TPS

Důležité je zmínit, že program TPS může pracovat ve dvou módech, a to ve *Standard mode*, který slouží pro práci a modelování v TPS a *Line Simulation mode*, který vytváří datový tok, tok materiálu, obsluhu signálů a řízení simulace. Přepínání módů je možné v záložce *Home* pod částí *Study*.

Simulaci lze řídit interně nebo externě. Přepínání řízení v TPS se provádí ve *File – Options – PLC*. Program TPS má tři možnosti chodu simulace, z toho jedna je interní a dvě externí (PLC).

CEE (Cyclic Event Evaluation) – Je základní interní nebo také off-line řízení simulace. V tomto módu je simulace ovlivňována pomocí logických bloků a modulů. Větvení operací se provádí pomocí *Sequence Editor*, kde se definuje takzvaný Ganttův

diagram. Diagram tvoří dílčí operace vázané v čase, které jsou spouštěny podmíněnými přechody.

Emulation – Toto řízení umožňuje čistě virtuální externí řízení buď přes SIMATIC nebo PLCSIM. Obě možnosti simulují vstupy a výstupy bez nutnosti reálného hardwaru.

External Connection – Využívá externí propojení TPS s reálným nebo simulovaným PLC. Pomocí reálných vstupů a výstupů lze přímo ovlivňovat chod simulace skrze program v PLC.

Při externím řízení nefungují modul objekty, ve kterých se dá částečně programovat. Všechny nezávislé funkce musí být implementovány pouze v logických blocích, které nejsou závislé na řízení PLC. [8]

V programu TPS jsou čtyři možnosti externího propojení:

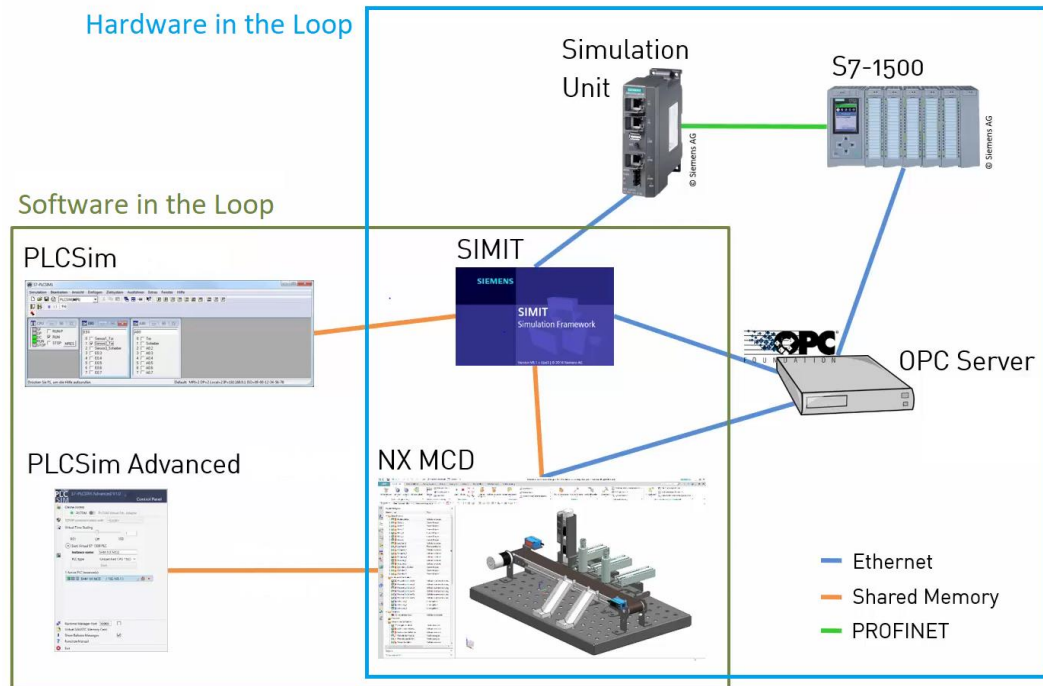
- OPC DA
- OPC UA
- Simulation Unit (SIMATIC)
- PLCSIM Advanced

2.4 Mechatronics Concept Designer

Mechatronics Concept Designer dále MCD, je modul CAD softwaru Siemens NX. Díky vlastnostem CAD softwaru je využit i pro prototypování buněk. Byl vytvořen pro 3D simulaci CNC obráběcích strojů, ale osvědčil se i pro ověřování koncepce větších výrobních jednotek. Výpočetní fyzikální jádro je založeno na přesném herním enginu Nvidia PhysX. Schopností tohoto jádra jsou i přesné simulace vzájemného silového působení těles. Simulace může být přehledně a snadno řízena z Ganttova diagramu nebo navázána na některý z externích zdrojů. Virtuální zprovoznění v MCD fyzikálně kopíruje reálný hardware. Těchto vlastností je možné využít při ověření návrhu, optimalizaci nebo virtuální výrobě. Výhodou integrace MCD v NX je podpora kooperativní práce na projektech. Práce inženýrů se paralelně dělí na mechanické, elektrické i programové části, na kterých mohou pracovat zároveň. NX s MCD je tak vhodné pro flexibilní a časově náročné projekty. [5]

2.4.1 Možnosti externího řízení MCD

Standardní způsoby externího řízení u MCD jsou totožné s programem TPS. Využití těchto komunikačních standardů v jednotlivých virtuálních metodách je vidět na obrázku Obr. 4. MCD dále definuje čtyři další možnosti externího řízení oproti TPS. [5]



Obr. 4 Možnosti propojení MCD [6]

Propojení s virtuálními jednotkami (SiL) může být realizováno se systémem SIMIT nebo softwarem pro emulaci PLC, PLCSIM Advanced. Pro tyto dvě možnosti MCD nabízí i možnost synchronizace reálného času. Další možností externího řízení bez nutnosti fyzického PLC je i propojení se softwarem Matlab Simulink. [6]

S fyzickým hardwarem je možné se spojit pěti způsoby (HiL). Nejčastější je použití M2M propojení přes OPC. MCD realizuje klientskou komunikaci s OPC serverem. Využívá obou standardů UA i DA. Další možnostmi jsou uživatelské protokoly TCP a UDP. Největším problémem UDP protokolu je jeho nízká spolehlivost. TCP je oproti tomu spolehlivý, ale pomalejší. Jedná se o asynchronní klient-server komunikaci, kde MCD zastává roli klienta. [6]

MCD nabízí možnost propojení Profinet. Toto propojení je založeno na SIMATIC NET standardně používaném pro průmyslové PC stanice. MCD je ve funkci Profinet master, který komunikuje pouze se zařízením *IO Device*. *IO Device* může být i PLC. Pokud by bylo takto PLC nastaveno, nevykonává se jeho program, ale vstupy a výstupy jsou ovlivňovány externě. Řízení je decentralizované přes Profinet z MCD. Využití takového propojení je při ověření řídicích rutin, které jsou naprogramovány v MCD, na reálném hardwaru. [6]

Možnosti externího propojení MCD:

- OPC DA
- OPC UA
- SHM (*Shared Memory Connection*) SIMIT
- PLCSIM Advanced
- Matlab Simulink
- TCP
- UDP
- Profinet

2.5 Porovnání MCD a TPS

Tato kapitola shrnuje zásadní výhody a nevýhody použití TPS a MCD v projektu Testad Barman. Nejedná se o výčet všech rozdílů, protože každý software je primárně určený na jiné aplikace. Oba produkty vyvíjí stejná firma, proto je očekávatelné, že novější MCD je v některých ohledech inspirováno funkcemi TPS. [5]

2.5.1 Propojení

V nástroji MCD je na výběr z osmi možností, což je dvojnásobný počet oproti TPS. Základní propojení je u obou softwarů stejné. Projekt sice využívá OPC UA nebo PLCSIM Advanced, které jsou dostupné u obou, ale pro budoucí využití ve výuce se zdá zajímavé propojení se Simulink Matlab. [6]

2.5.2 Flexibilita

MCD je jako doplňkový modul integrován do prostředí NX, které má plnou podporu pro tvorbu CAD modelů. Výhoda plynoucí z této integrace je odstranění nutnosti převádět modely přes konverzní formáty (*JT*), jako je tomu u TPS. Tato výhoda umožňuje provádět změny v konstrukci okamžitě po odhalení v simulaci. To zrychluje efektivitu práce a snižuje čas potřebný na prototypovou výrobu. [5]

V prostředí TPS je možné tvořit jednoduché logické nebo matematické funkce, ale jejich implementace je složitá a neobratná. Komplexnější funkce mohou být realizovány skrze OLP příkazy v sekvenčním diagramu. MCD má tyto logické a matematické funkce optimalizovány pro přehlednější a snazší implementaci. Komplexnější závislosti dílů lze vytvářet v jazyku C#. [5]

V TPS je možné definovat pouze dva druhy spojů *Revolute* (Rotační spoj) a *Prismatic* (Lineární posuv). Oproti tomu MCD má široký výběr spojů, které se stále rozšiřují. Příkladem může být šroubový spoj, jehož tvorba v TPS by zabrala daleko více času. [5]

2.5.3 Robotické simulace

Modul MCD není přímo určen k virtualizaci celých robotických linek, i když je toho částečně schopen. NX 12 nabízí propojení s externími moduly pro programování robotů. Primární aplikační působnost tohoto nástroje je zaměřena na ověření strojů vlastní konstrukce. [5]

Výhodou TPS zůstává snadná implementace robotických pracovišť, ale za určitých podmínek. Podmínkou pro využití těchto nástrojů je virtuální kontrolér robota. Jedná se o programovou emulaci kinematických výpočtů fyzického kontroléru. Tím je docílena i snadná konverze robotického programu se simulací. Výrobci robotů u nových strojů tuto softwarovou podporu poskytují. [5]

Bohužel projekt Barman disponuje robotem, u kterého tato podmínka není splněna. Robot byl vyroben ještě v době, kdy TPS nebo jiné virtualizační nástroje neexistovaly. Domluvou tedy bylo, že robotický manipulátor je nahraditelnou součástí stroje Barman. Díky tomu, jsme ustoupili od implementace do TPS, ale zaměřili pozornost na MCD. Tato změna proběhla v průběhu roku, a proto se část praktické práce věnuje i TPS.

NX ve verzi 1903 obsahuje nové funkce pro robotická řešení v MCD. Nově nabízí funkci inverzní kinematiky, a to s možností online nebo offline programování. Dále modul nabízí funkci *Envelope* (Obálka), která vypočítá pracovní prostor robota nebo jiné kinematiky. Při aplikaci většího počtu robotů je možné tyto prostory sledovat funkcí *Interference* (Kolize obálek). Tyto nové funkce zasahují do oblasti působnosti TPS. Je možné, že by MCD, respektive NX, mohlo nahradit starší produkty značky Siemens.

2.6 SIMIT

Je nástroj firmy Siemens pro simulaci technologických a řídicích systémů v reálném čase. Využití programu je zasazeno v konceptu virtualizace. Přímé propojení MCD a SIMIT využívá sdílené paměti. Proto je tak možné využít synchronizace obou simulačních nástrojů. [5]

K propojení SIMIT s fyzickým PLC je zapotřebí hardwarové podpory od společnosti Siemens. Jednotka SimBox nebo nově Simulation Unit je hardwarové komunikační rozhraní, které používá protokol PROFINET. Výhodou může být možnost připojení simulovaného PLC pomocí programu PLCSIM. PLCSIM není závislé pouze na simulaci řady PLC S7-1500, ale i dalších typů PLC jako například S7-1200. [6]

2.7 OPC UA

Jedná se o komunikační standard M2M (Machine to machine) nezávislý na platformě, vyvinutý firmou OPC Foundation. Zkratka UA znamená „Unified Architecture“ neboli sjednocená architektura. Původní architektura OPC byla určena pouze pro operační systémy firmy Microsoft a na jiných platformách ji nebylo možné použít. OPC UA využívá obecných komunikačních standardů jako jsou TCP/IP, HTTP nebo SOAP, díky kterým je možné OPC UA využít i v PLC automatech. [7]

Vlastnosti protokolu UA oproti DA jsou:

- Nezávislost na platformě
- Škálovatelnost
- Bezpečnost založená na nových standardech
- Nastavitelný time-out pro každou službu
- Rozdělení objemných datagramů na menší části

Výhodou UA oproti DA (*Data Access*) je, že nedefinuje konkrétní přístupy k datům, ale pouze formát zpráv. Tím je docíleno zjednodušení oproti jiným OPC standardům, které nedokážou přenášet procesní data, alarmy nebo historická data bez nutnosti změny přístupu. OPC DA je starší standard, který definuje i přístupy k různým datovým typům. Nevýhodou staršího standardu DA je i závislost na platformě MS Windows. OPC UA komunikace podporuje dva protokoly. Rozdíl v aplikaci pro integrátory je pouze v URL OPC serveru. [7]

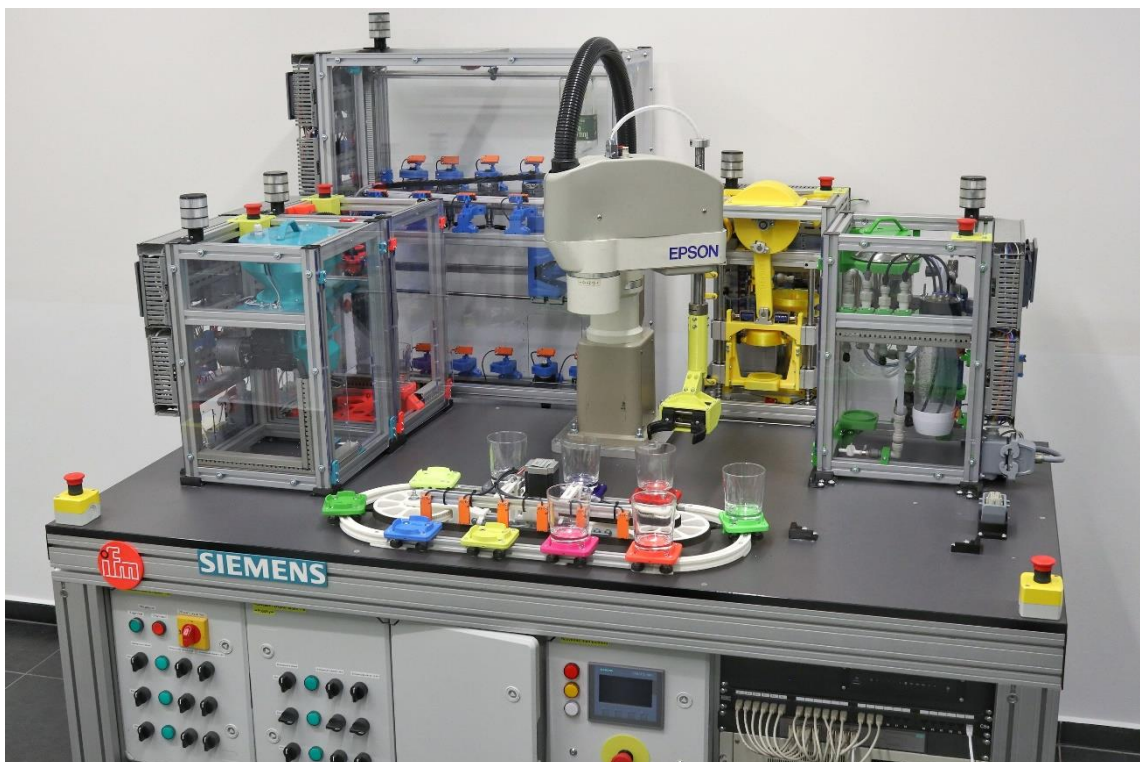
- `opc.tcp://Server` (binární protokol)
- `http://Server` (webová služba)

Bezpečnost komunikace je zajištěna pomocí autentizace, autorizace, šifrování a certifikáty. MCD v aktuální verzi podporuje pouze propojení bez zabezpečení.

2.8 TestBed Barman

Tato část má za úkol seznámení se s projektem Barman a jeho aktuálním stavem. Jedním z bodů praktické části je i výběr vhodné buňky a možnosti její virtualizace v prostředí TPS nebo MCD. Barman je rozdělen do osmi autonomních buněk, z toho dvě jsou transportní a šest výrobních. Buňky jsou umístěny na pracovním stole v takzvaných slotech. Každý slot je vybaven dvěma elektrickými konektory. První konektor slouží pro připojení napájecí sběrnice a druhý pro komunikaci. Komunikační rozhraní využívá standardu profinet. Z neelektrických konektorů je buňka také vybavena rozhraním pro chladicí okruh. Buňky jsou standardizovány do dvou rozměrových formátů, malé buňky (š: 330 mm × h: 330 mm × v: 500 mm) a velké buňky (š: 760 mm × h: 330 mm × v: 1500 mm). Na stole jsou čtyři sloty pro buňky malého rozměru, za stolem jsou dvě místa, a to na velkou nebo malou buňku. Malá buňka je za stolem umístěna na expanzní stoličce, která je připevněna z boku zadní strany stolu. Umístění slotů je závislé na dosahu manipulačního robota. [11]

Buňky mají jednotné výdejné místo pro skleničky. Stání pro skleničky jsou vybavena chytrou RFID čtečkou. Čtečka by měla v budoucnu sloužit k identifikaci sklenic, čtení receptury a naplánování výroby. Tím bude dosaženo maximální variability výroby. Součástí informací na tomto čipu bude i receptura a stav, v jakém se receptura nachází. Díky chytrým čtečkám bude vyjednána cesta k budoucí operaci receptury. Využitím IoT principů bude rozhodování a plánování výroby zcela decentralní. [11]



Obr. 5 TestBed Barman

2.8.1 Dopravník

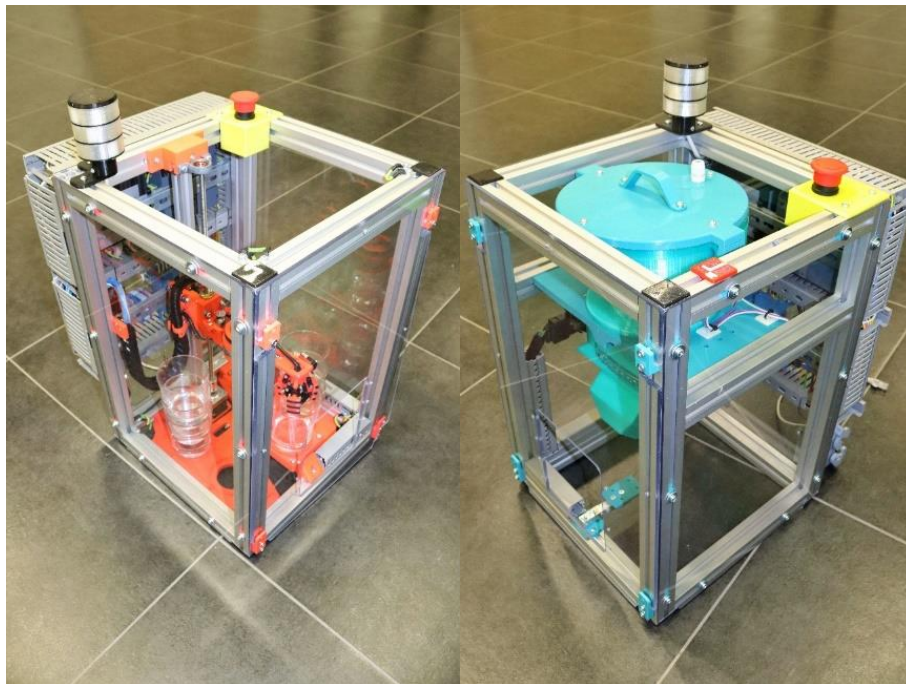
Dopravník je člen, který propojuje stroj s veřejností, popřípadě s obsluhou. Jeho hlavním úkolem je transportovat připravené nápoje mimo operační prostor robota. Díky dopravníku je možné zabezpečit částečnou bezpečnost stroje Barman. Další schopností dopravníku je i přechodné uložení rozpracovaného nápoje, nebo odběr prázdných (vypitých) sklenic. Dopravník je vybaven deseti vozíky na sklenice a je schopen detekovat stav i obsah sklenic. [11]

2.8.2 Manipulátor

Jednotka zajišťuje transport skleniček mezi buňkami a dopravníkem. Jako transportní jednotka je použit SCARA robot EPSON H554BN vybavený efektoem schopným uchopit sklenici. Robot je řízen kontrolérem EPSON SRC 310, který využívá pro programování jazyk SPEL 3. Interface pro kontrolér je RS232, skrze který je robot ovládán z nadřazeného PLC. [11]

2.8.3 Sklad sklenic

Buňka skladu sklenic je malého formátu. Její funkcí je vydávat čisté sklenice a ukládat špinavé sklenice. K tomu slouží dvouosý manipulátor a čtyři vertikální zásobníky, ve kterých se sklenice stohují na sebe. Manipulátor je umístěn tak, aby pokryl svým pohybem dva sklady na levé straně, dva sklady na pravé straně a výdejní místo umístěné v ose manipulátoru. [11]



Obr. 6 Buňka sklad sklenic vlevo, buňka drtič ledu vpravo

2.8.4 Drtič ledu

Buňka drtiče ledu slouží ke skladování, drcení a nadávkování ledu do sklenice. Led se skladuje ve formě kostek v izolovaném zásobníku nad drtičem. Drtič je nožový rotační. Buňka dávkuje požadované množství ledu do sklenice pomocí tenzometrického nosníku na výdejním místě. [11]

2.8.5 Zásobník nealkoholických nápojů

Buňka se skládá ze čtyř tanků na nealkoholické nápoje. Tyto tanky jsou chlazeny z chladicího okruhu stolu. Buňka je určena pro dávkování většího množství tekutin jako jsou koly, toniky nebo džusy. Dávkování tekutin je zprostředkováno pomocí solenoidových ventilů a pulsních průtokoměrů. Sklenice je vážena tenzometrickým mostem, který snímá nadávkované množství. Zatím nedokončená buňka. [11]

2.8.6 Zásobník alkoholických nápojů

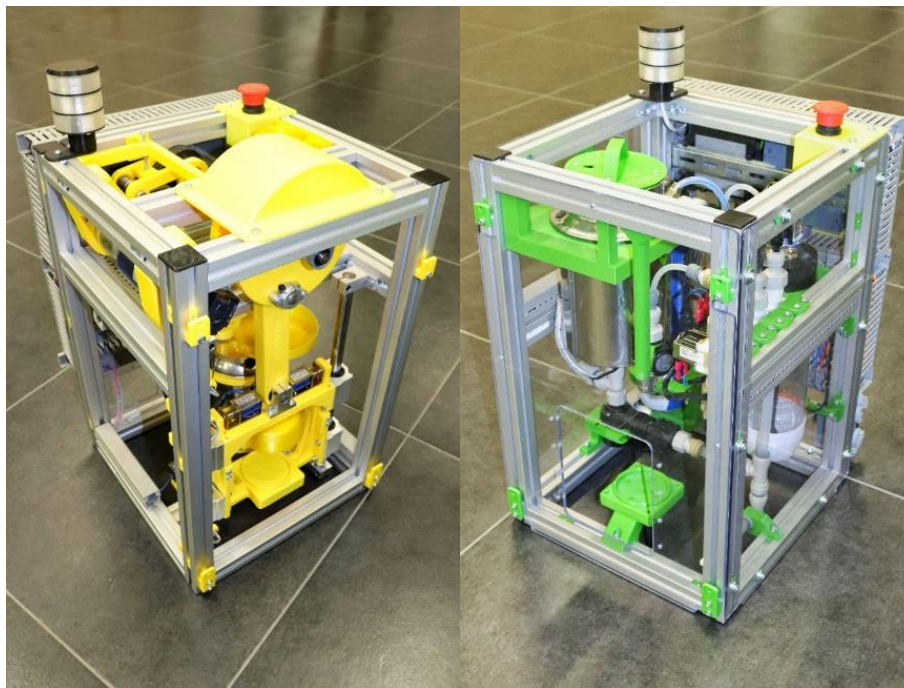
Tato buňka má nejdůležitější úkol ze všech. Pomocí lineárního manipulátoru s obručí na rotačním kloubu manipuluje se sklenicí. Díky tomu je buňka schopna do sklenice dávkovat tekutiny z vybraného zásobníku. Může obsloužit až dvacet zásobníků na tekutinu. Každý zásobník je vybaven mechanickým gravitačním dávkovačem. [11]



Obr. 7 Buňka sklad alkoholických nápojů

2.8.7 Šejkr

Tato buňka má za úkol protřepat obsah sklenice. Šejkr je poháněn stejnosměrným motorem a pracuje na principu klikové hřídele, který převádí pohyb rotační na lineární. Buňka nemění obsah skleničky, ale je nutné simulovat pohyb šejkru. [11]



Obr. 8 Buňka šejkr vlevo, buňka sodobar vpravo

2.8.8 Sodobar

Buňka skladuje pitnou vodu a vytváří z ní nasycenou sodu pomocí CO_2 . Sodovka se vyrábí v dávkách, kde velikost dávky se odvíjí od velikosti sytící lahve od firmy *SodaStream*. Po vytvoření jedné dávky sody se obsah dávkuje do sklenic opět pomocí tenzometrického mostu, který váží sklenici. [11]

Jako u buňky drtiče ledu, tato buňka nemá žádné pohyblivé části, pouze generuje materiál do sklenice. Simulace plnění bude probíhat jako „montáž“ sklenice a jejího obsahu. [11]

3. VIRTUALIZACE

V rámci semestrální práce bylo cílem využít program TPS 14. Zjistit možnosti virtualizace stroje Barman a využít k tomu prací, které se TPS již v minulosti zabývaly. V té době nebylo možné modul MCD využít. Se získáním licence na rozšíření NX o modul MCD, bylo konzultováno jeho využití. Výhody použití MCD jsou shrnuty v kapitole 2.5.

V této části je cíleno na výběr vhodného modelu a jeho virtuální zprovoznění. Kapitola cituje CAD modely, které jsou převzaty z projektu *Barman* a dále upravovány. Modely byly v rozpracovaném stavu některé více, jiné méně kompletní. Pro potřeby této práce byla použita CAD/CAM/CAE softwarová platforma NX od společnosti Siemens PLM. Tento softwarový balíček obsahuje i vybavení pro tvorbu simulace a její validace ve formě virtuálních dvojčat. Balíček se jmenuje *Mechatronics Concept Designer*.

Zadání diplomové práce bylo realizováno ve starší verzi softwaru NX 12.0 v českém překladu. Ve finální fázi práce došlo k migraci projektu na novější aktualizaci NX 1903 v anglické verzi. Aktualizace nastala z důvodu autorova zapojení v jiném projektu. Bylo doporučeno odinstalovat stávající verzi, kvůli nestabilitě. Při současně instalovaných verzích docházelo k problémům. Číslování softwarových iterací je poněkud zavádějící, protože u značky NX se můžeme dopátrat k označení NX 19.03 nebo NX 1903.

3.1 Práce v TPS

V první části práce byla vybrána pro virtualizaci jednotka manipulátoru. Simulace je uskutečněna v programu TPS verze 14, který byl popsán v teoretické části výše, kapitola 2.3. V této kapitole bude prezentována práce s modely v prostředí TPS, jejich oživení a tvorba robotických programů. V neposlední řadě propojení programů s řídicím systémem. [12][13][14][15][16]

3.1.1 Modely

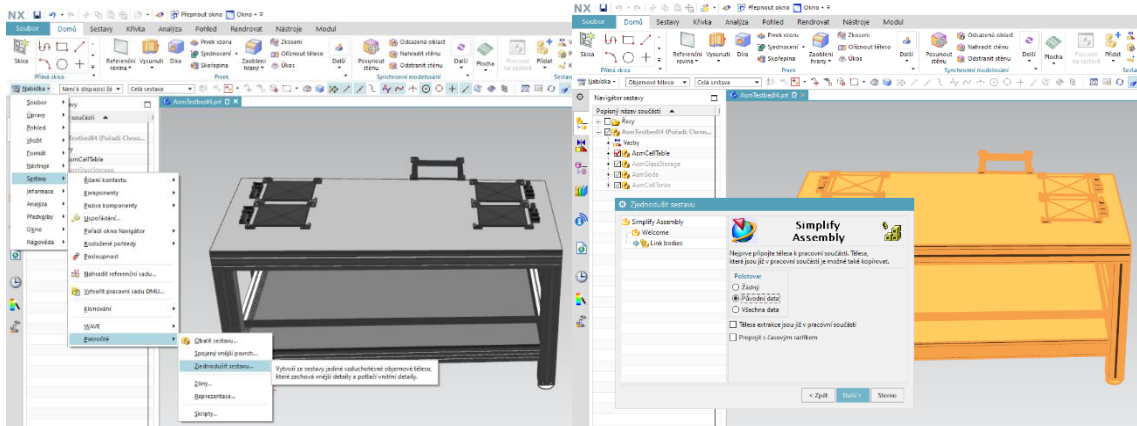
Většina modelů byla tvořena v CAD softwaru NX od firmy Siemens. Pouze model velké buňky je nakreslen v programu SolidWorks a bylo jej nutné uložit do formátu STEP, který je možné otevřít i v programu NX.

V prostředí NX je vhodné nejprve sestavy zjednodušit a poté exportovat do formátu JT, pro import do programu TPS. V této kapitole bude nastíněn postup převodu formátů na příkladu stolu.

Modely robota a malých buněk byly převzaty z bakalářské práce Bc. Davida Buchala a upraveny. Tyto modely jsou ve formátu COJT, který je generovaný programem TPS. [12]


3.1.1.1 Stůl

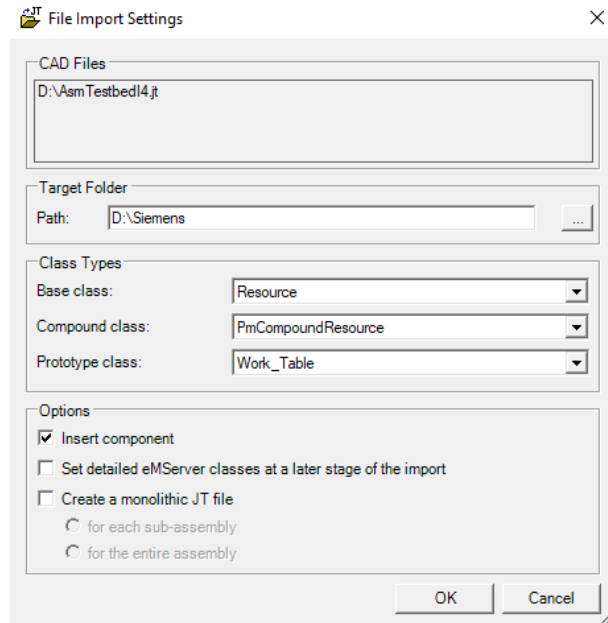
Stůl je základním kamenem stroje Barman a udává konfiguraci a koncepci celému zařízení. Použitý model stolu není přesnou kopií reálného stolu, ale obsahuje nezbytné části pro virtualizaci stroje. Například v modelu není vymodelovaný hardware pod pracovní deskou. V reálném zařízení je zde umístěna spousta nepostradatelných částí, jako elektrický rozvaděč, kontrolér k robotu, chladicí jednotka a síťové prvky. Tyto prvky ale nemají v simulaci funkční význam.



Obr. 9 Funkce zjednodušení sestavy v NX

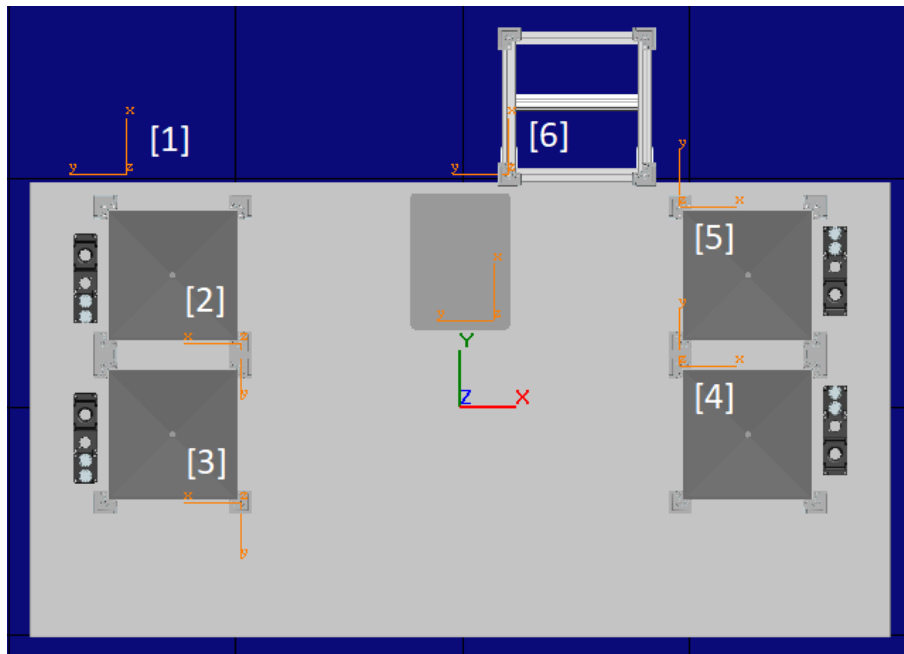
V programu NX je stůl reprezentován sestavou dílů. Pokud provedeme export do formátu JT, vytvoříme soubor pro všechny součásti. Zjednodušení sestavy je pokročilá funkce v NX. Aplikací funkce a následným exportem sestavy získáme jeden soubor reprezentující sestavu. Zjednodušený model se nevloží po dílech a neznehlední projekt. Jednotlivé díly jsou ale stále dosažitelné v *Set Modeling Scope*, viz další kapitoly. Postup zjednodušení sestavy je delší, ale v každém kroku jsme instruováni, jak máme pokračovat.

V programu TPS, záložce *File – Import/Export* vybereme  *Convert and Insert CAD Files*. V otevřeném okně stejného názvu přidáme příslušný soubor s tlačítkem *Add...*. Po vybrání souboru ve formátu JT se zobrazí okno *File Import Settings*, kde vybereme možnosti dle obrázku Obr. 10. Pro další modely bude postup stejný, jen v oblasti *Prototype Class* vybereme jinou funkci, například pro model efektoru *Gripper*.



Obr. 10 Nastavení importu modelu

Stůl má definovány body *frames* (rámy) pro rychlé pokládání a orientaci buněk. To urychlí úpravu rozmístění buněk. Buňky je tak možné snadno přemístit dle aktuální konfigurace stroje. Tyto body se přidají tak, že model otevřeme v *Set Modeling Scope*, vytvoříme *Create Frames*. Před ukončením *End Modeling* je důležité u každého rámu zvolit *Set Object to be Preserved*, aby byly viditelné i po ukončení editace modelu a daly se použít pro pokládání buněk.



Obr. 11 Souřadnice buněk na stole v TPS



Tab. 3 Souřadnice buněk v TPS

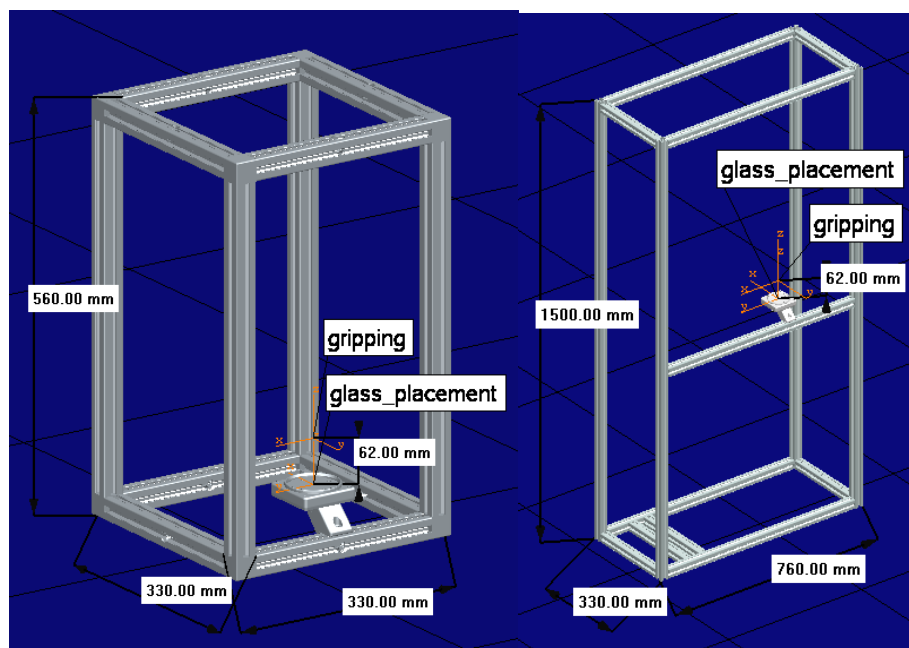
Buňka č.	X [mm]	Y [mm]	Z [mm]	Rx [°]	Ry [°]	Rz [°]
1	-900	-455	-968	0	0	90
2	-485	145	8	0	0	180
3	-485	-210	8	0	0	180
4	458	90	8	0	0	0
5	485	440	8	0	0	0
6	107,5	510	8	0	0	90
Podstava	0	320	0	-90	0	0

3.1.1.2 Buňky

Z buněk jsou importovány pouze jejich rámy s výdejním místem skleniček. Místo pro uchopení sklenice gripperem je pouze pro zrychlení tvorby robotických operací.

Pro velkou buňku nebyl vytvořen model v konstrukčním programu NX, ale v programu Solidworks 2018. Bylo proto nutné jej převést nejprve do formátu STEP. Tento formát je schopen otevřít i program NX. Následně je z NX exportován do potřebného JT formátu jako u ostatních modelů.


Buňky umístíme na stůl do předem připravených rámu nebo podle souřadnic v tabulce Tab. 3. Po označení buňky klikem pravého tlačítka zobrazíme kontextové menu. Vybereme  *Relocate*, klikneme na příslušný rám na stole a potvrdíme tlačítkem *Apply*. Pokud bychom potřebovali změnit pozici mimo rámy, je možné v části *To frame*: stisknout tlačítko  *Create Frame of Reference* a vyplnit souřadnice ručně.

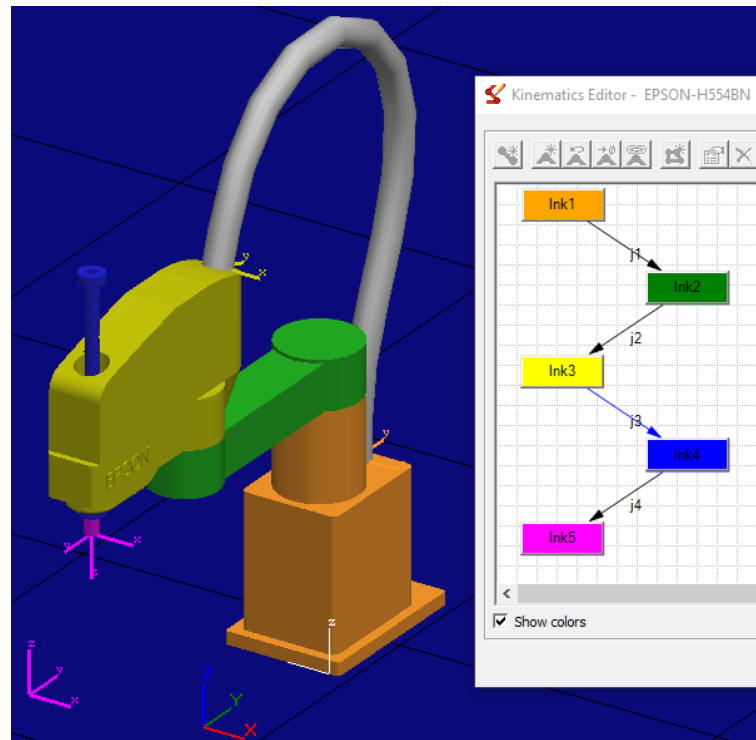


Obr. 12 V levé části model malé buňky, v pravé části model velké buňky


3.1.2 Robot



Tato část volně navazuje na bakalářskou práci Bc. Davida Buchala, který vytvářel model robota a jeho kinematiku.

Vložení robota se provádí v záložce *Modeling – Components* tlačítkem  *Insert Component*. Vybereme model „*epson-h554bn.cojt*“ a otevřeme ho. Podle návodu v citovaném zdroji [12] vytvoříme kabel robota.





Obr. 13 Model robota EPSON


Důležitým krokem je definice kontroléru. To se provádí v záložce *Robot – Setup*. Tlačítkem  *Controller Setting* vyvoláme stejnojmenné menu. Ve sloupci *Controller* vybereme možnost *default*. Jelikož nemáme virtuální kontrolér pro tento typ robota, je defaultní nastavení jediná možnost. Výrobce u tohoto robota neposkytuje soubory pro virtuální kontrolér.

Podmínkou pro externí řízení takového kontroléru jsou signály. Ty budou řídit defaultní robot kontrolér. V záložce *Control – Robot* stiskneme  *Create Robot Start Signals*. V té samé části lze otevřít okno editoru robotických signálů tlačítkem  *Robot signals*. Zde je možné dále editovat signály. Pokud již máme nastavené externí řízení, je zde možné definovat i adresu a externí připojení. Viz kapitola 3.1.4.

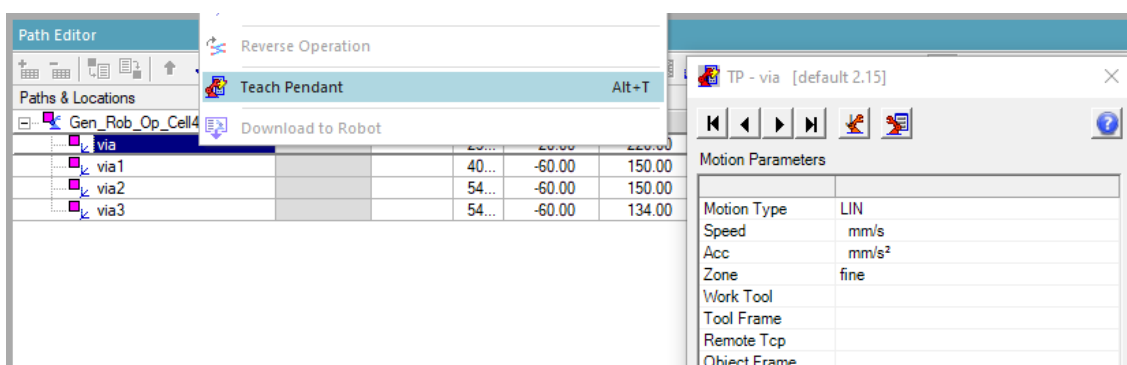
PLC Signal Name	Robot Signal Name	I/O	Signal Function	HW Type	Address	External Connection
"EPSON-H554BN_startProgram"	startProgram	Q	Starting Program	BOOL	1.0	OPC_POKUS
"EPSON-H554BN_programNumber"	programNumber	Q	Program Number	BYTE	1	OPC_POKUS
"EPSON-H554BN_emergencyStop"	emergencyStop	Q	Program Emergency Stop	BOOL	2.0	OPC_POKUS
"EPSON-H554BN_programPause"	programPause	Q	Program Pause	BOOL	3.0	OPC_POKUS
"EPSON-H554BN_programEnded"	programEnded	I	Ending Program	BOOL	4.0	OPC_POKUS
"EPSON-H554BN_mirrorProgramNumber"	mirrorProgramNumber	I	Mirror Program Number	BYTE	2	OPC_POKUS
"EPSON-H554BN_errorProgramNumber"	errorProgramNumber	I	Error Program Number	BOOL	5.0	OPC_POKUS
"EPSON-H554BN_robotReady"	robotReady	I	Robot Ready	BOOL	6.0	OPC_POKUS

Obr. 14 Signály robota






Robotické operace se vytváří v záložce *Operation – Create Operation*. Tlačítkem  *New Operation* se otevře rolovací nabídka s operacemi. Pro simulaci pohybu robota je vhodné použít operace  *Create Generic Robotic Operation*. Trajektorie a její body vkládáme pomocí nástrojů v nabídce *Add Location*. Tyto operace pouze simulují trajektorii robota bez práce s gripprem nebo jiným nástrojem. Práce s nástrojem se dá dodatečně definovat buď pomocí OLP příkazů nebo jako je to v tomto případě externím řízením gripperu signály z PLC.

Jednotlivé operace můžeme sloučit do sdružených operací takzvaných *Compound Operation*. Ty slouží k organizaci operací v *Operation Tree*. Vytvoříme je stejně jako standardní operace výběrem  *New Compound Operation*.

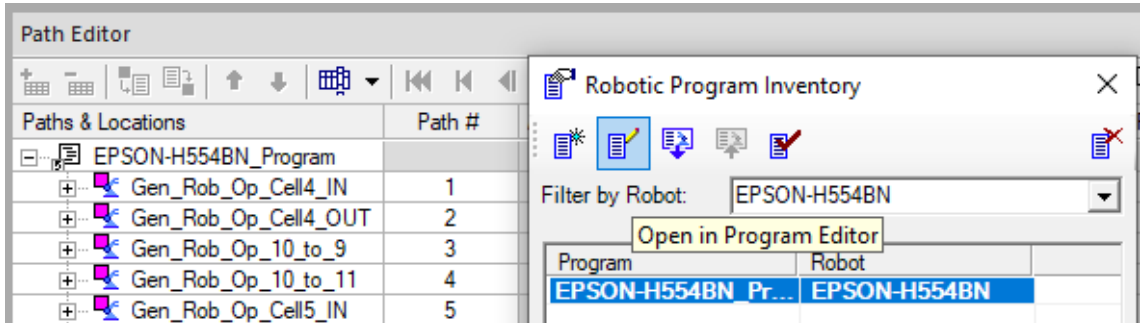
Chceme-li, aby byl pohyb robota mezi body lineární, je nutné jednotlivé body (*vias*) editovat. Operaci tažením myši přidáme do *Path Editoru* z *Operation Tree*. Poté pravým tlačítkem přes menu otevřeme *Tech Pendant*, kde se změní typ pohybu (*Motion Type*) na *LIN* viz Obr. 15.



Obr. 15 Editace bodů operace

Robotický program se skládá z robotických operací. Z okna *Operation Tree* vybereme všechny *Compound* operace, které chceme mít v novém programu. V záložce *Robot – Program* vybereme  *Create Robotic Program from Compound Operation*. Tím se vytvoří robotický program. Editace nového robotického programu je možná v okně  *Robotic Program Inventory*. Označením žádaného programu a stiskem  *Open in Program Editor* se program otevře v okně *Path Editor*, kde je možné přidávat  nebo odebírat  jednotlivé robotické operace a editovat *Path#* (cestu) jednotlivých

robotických operací. Každá robotická operace musí mít unikátní číslo cesty, kterým je spouštěna. Výběr spouštěné operace probíhá skrze externí signál „*Program number*“. Díky tomu jsme schopni řídit simulovaný kontrolér z PLC nebo jiného externího zdroje.

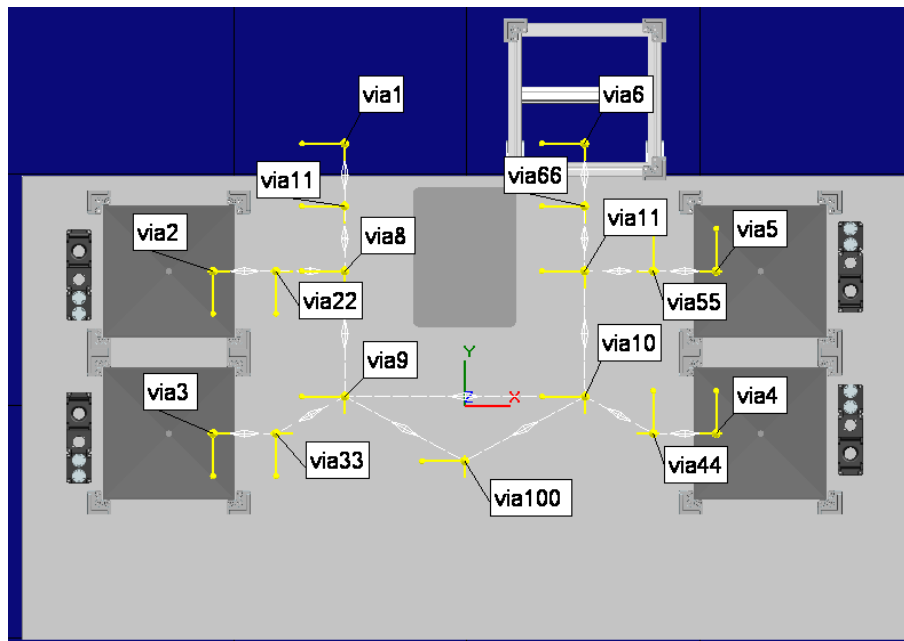


Obr. 16 Editace robotického programu

Robot, který má kontrolér shodný s virtuálním, nám značně zjednoduší práci s implementací robotického programu. U takového robota je možné robotický program i s robotickými operacemi jednoduše přenášet mezi simulací a reálným hardwarem.

Při tvorbě robotických operací jsem vycházel z robotického programu fyzického robota, který vytvářel ve své diplomové práci Ing. Tomáš Sýkora. Všechny body jsem neimplementoval, ale snažil jsem se je zjednodušit. Program není přenositelný mezi simulací a robotem, jelikož neexistuje virtuální kontrolér. [13]

Jak je vidět na obrázku Obr. 17, bodů je zde více, než je popsáno v tabulce Tab. 4, ale tyto body jsou pouze průjezdné a v řízení nejsou podstatné. Definují pouze trajektorii operace. Důležité body jsou koncové viz tabulka Tab. 4.



Obr. 17 Náhled na body a operace robota

Tab. 4 Koncové body operací

Bod	X [mm]	Y [mm]	Z [mm]	Rx [°]	Ry [°]	Rz [°]	Poznámka
Via1	-257,5	565	134	0	0	180	Buňka 1
Via2	-540	290	134	0	0	-90	Buňka 2
Via3	-540	-60	134	0	0	-90	Buňka 3
Via4	540	-60	134	0	0	90	Buňka 4
Via5	540	290	134	0	0	90	Buňka 5
Via6	257,5	565	134	0	0	180	Buňka 6
-	-	-	-	-	-	-	-
Via8	-257,5	290	220	0	0	180	Uzel - 1, 2
Via9	-257,5	20	220	0	0	180	Uzel - 3
Via10	257,5	20	220	0	0	180	Uzel - 4
Via11	257,5	290	220	0	0	180	Uzel - 5, 6

Via 1-6 jsou body v buňkách, via 8-11 jsou uzlové body (rozcestí) a via 100 je bod zvratu robota. Pokud robot projíždí bodem 100, mění se mu orientace. Orientací se myslí, že rameno robota je zlomené vlevo nebo vpravo.


Zatím chybí obsluha dopravníku, pro kterou byl vyhrazen neimplementovaný bod via 7. Nejspíš tento bod nebude koncový, ale jen čekací. V tomto bodu se robot zastaví a pro pokračování robotické operace bude čekat na synchronizační signál. Toto vyčkání by se nejnadhěji implementovalo v TPS pomocí OLP příkazů. Zatím je ponechána možnost externí synchronizace mimo TPS. Program 19 a 20 je přejezd z uzlu 9 do 10 a naopak s reverzací orientace robota.


Tab. 5 Robotické programy v TPS

Číslo prog.	Start	Cíl	Číslo prog.	Start	Cíl
1	Uzel 10	Buňka 4	11	Uzel 9	Uzel 8
2	Buňka 4	Uzel 10	12	Uzel 9	Buňka 3
3	Uzel 10	Uzel 9	13	Buňka 3	Uzel 9
4	Uzel 10	Uzel 11	14	Uzel 8	Uzel 9
5	Uzel 11	Buňka 5	15	Uzel 8	Buňka 2
6	Buňka 5	Uzel 11	16	Buňka 2	Uzel 8
7	Uzel 11	Buňka 6	17	Uzel 8	Buňka 1
8	Buňka 6	Uzel 11	18	Buňka 1	Uzel 8
9	Uzel 11	Uzel 10	19	Uzel 10	Uzel 9
10	Uzel 9	Uzel 10	20	Uzel 9	Uzel 10

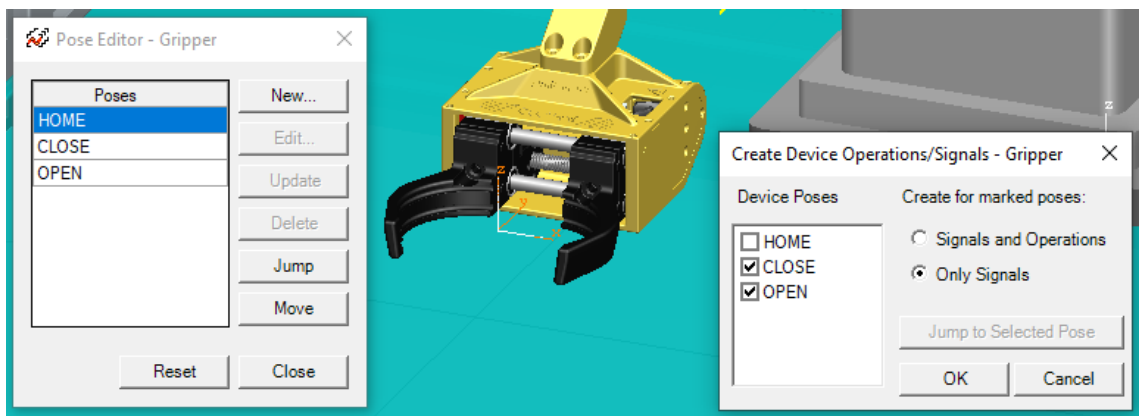
3.1.3 Efektor

Efektor, nebo častěji nazývaný gripper, je nástroj robota. Stejně jako člověk je i robot bez nástroje nepoužitelný. Základní funkcí tohoto nástroje je být v kontaktu s manipulovaným předmětem (sklenicí). V našem případě se jedná o gripper se dvěma pohyblivými chapadly. Chapadla působí silou proti skleničce a tím ji fixují. V průběhu této práce procházel gripper fyzickou změnou. Proto je v simulaci používán model V.2. Třetí verze má stejné rozměry, pouze není vidět převodový mechanismus. Navíc je v chapadle umístěn mikrosplínač, který detekuje přítomnost sklenice.

V programu TPS se efektor vloží s definicí *Gripper*. Pokud bychom ho definovali jinak, nebylo by ho možné připevnit na *Tool Frame* robota. Aby bylo možné gripper řídit externě, jsou mu definovány řídicí signály. Pro úspěšné vytvoření signálů je nejprve potřeba správně definovat kinematiku gripperu. Dále je nutné definovat pozice otevřeno a zavřeno. Tyto pozice se definují ve funkci  *Pose editor*. Nejprve vytvoříme pozici OPEN, a to stisknutím tlačítka *New* a nastavením gripperu na fyzický limit plného otevření *Value = 0*. Druhá pozice CLOSE se vytvoří obdobně, jen se gripper nastaví na *Value* takové, aby sevřel sklenici. V tomto případě to je 12,11 mm. Parametr je možné získat experimentálně pomocí ručně umístěné sklenice v gripperu.

Signály efektoru se generují pomocí nástroje  *Create Device Operations/Signals*. Vybereme pouze polohy OPEN a CLOSE a z možností generování vybereme „Only Signals“.

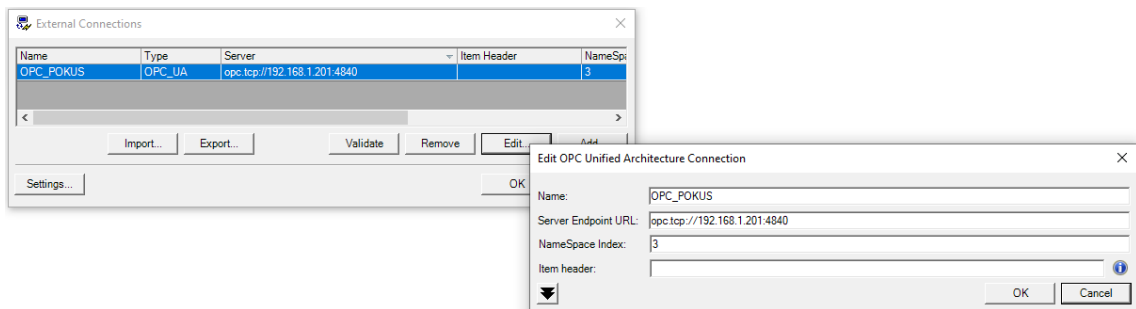
Program TPS defaultně definuje čtyři signály pro gripper. Jsou to dva výstupní signály (snímače) „*Gripper_at_open*“, „*Gripper_at_close*“, které indikují aktuální pozici gripperu a dva vstupní (řídicí) signály „*Gripper_to_open*“, „*Gripper_to_close*“, kterými jsme schopni externě ovládat otevření nebo zavření efektoru. [12]



Obr. 18 Vytvoření signálů efektoru

3.1.4 Konfigurace externího připojení

Při testování robotické simulace bylo využito externího řízení z fyzického PLC. Propojení bylo realizováno pomocí protokolu OPC UA. Nejprve je nutné mít funkční OPC server vytvořený v PLC S7-1500. Klientská konfigurace v TPS se provádí v menu *File – Option – PLC*. Vybereme možnost *External Connection*.



Obr. 19 Nastavení externího spojení

Poté je nutné namapovat signály v TPS. To se provádí v okně *Signal Viewer*, které je dostupné z rolovacího menu na záložce *View – Screen Layout – Viewers*. U každého ze signálů, které chceme používat externě, zaškrtneme checkbox *PLC Connection* a z položky *External Connection* vybereme námi nadefinovanou OPC komunikaci. Dalším nastavením je přiřazení adresy signálů. To je potřeba, pokud bychom použili přímé propojení s PLCSIM Advanced. Tím by se signály paměťově namapovaly k vstupům virtuálního PLC. Pro komunikaci s OPC si adresy zvolíme libovolně.

Poznámka:

Všechny signály, které jsou spojeny přes OPC musí mít název v uvozovkách! Názvy je nutné ručně upravit, aby OPC server v PLC dokázal správně namapovat proměnné. Toto platí pouze u signálů v TPS. V PLC jsou tyto signály bez uvozovek.

Signal Name	Memory	Type	Address	IEC Format	PLC Connection	External Connection	Resource	Comment
"EPSON-H554BN_programEnded"	<input type="checkbox"/>	BOOL	4.0	I4.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_mirrorProgramNumber"	<input type="checkbox"/>	BYTE	2	I2	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_errorProgramNumber"	<input type="checkbox"/>	BOOL	5.0	I5.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_robotReady"	<input type="checkbox"/>	BOOL	6.0	I6.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_startProgram"	<input type="checkbox"/>	BOOL	1.0	Q1.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_programNumber"	<input type="checkbox"/>	BYTE	1	Q1	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_emergencyStop"	<input type="checkbox"/>	BOOL	2.0	Q2.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"EPSON-H554BN_programPause"	<input type="checkbox"/>	BOOL	3.0	Q3.0	<input checked="" type="checkbox"/>	OPC_POKUS	● EPSON-H554BN	
"Gripper_at_CLOSE"	<input type="checkbox"/>	BOOL	13.0	I13.0	<input checked="" type="checkbox"/>	OPC_POKUS	● Gripper	
"Gripper_at_OPEN"	<input type="checkbox"/>	BOOL	12.0	I12.0	<input checked="" type="checkbox"/>	OPC_POKUS	● Gripper	
"Gripper_to_CLOSE"	<input type="checkbox"/>	BOOL	11.0	Q11.0	<input checked="" type="checkbox"/>	OPC_POKUS	● Gripper	
"Gripper_to_OPEN"	<input type="checkbox"/>	BOOL	10.0	Q10.0	<input checked="" type="checkbox"/>	OPC_POKUS	● Gripper	



Obr. 20 Signály potřebné k simulaci

3.2 Práce v MCD

Jako demonstrátora pohybových vlastností byla vybrána výrobní buňka Šejkr (*Shaker*). Tato buňka provádí operaci míchání nápoje a následně se provede čištění. V této buňce nedochází ke změně obsahu sklenice. Kinematika buňky se jevila výhodná, jak po vizuální stránce, tak vlastnostmi. Nacházejí se zde základní kinematické spoje, kterými jsou rotace a posuv. Proto je tato buňka vhodná pro demonstraci nástroje MCD.

Virtuální model buňky Shaker v prostředí MCD definuje virtuální vstupy i výstupy odpovídající reálným. Obecně je možné simulovat a snímat fyzikální veličiny jako poloha, rychlost a vibrace. Pomocí virtuálních aktuátorů lze v modelu ovlivňovat polohu, rychlost, sílu nebo moment jednotlivých vazeb. Celkově se simulace skládá z hmotných součástí, které jsou spojeny pomocí kinematických vazeb. Simulují se vnitřní i vnější síly, jako gravitace působící na součásti buňky, kolize pružné i nepružné, nebo tření mezi kolidujícími součástmi. [17][18][19]

3.2.1 Struktura projektu


Projekt v programu MCD je strukturalizován do osmi základních částí. Do stromové struktury se dostaneme pomocí záložky *Physic navigator*  (Navigátor fyzikálních vlastností). Všechny MCD vlastnosti se realizují v pracovní sestavě, popřípadě pracovním dílu. Změnu pracovní sestavy provedeme tlačítkem  *Make Work Part* po označení dílu nebo sestavy.

Name	Type
+ Basic Physics	
+ Joints and Constraints	
+ Materials	
+ Couplers	
+ Sensors and Actuators	
+ Runtime Behaviors	
+ Signals	
+ Signal Connection	

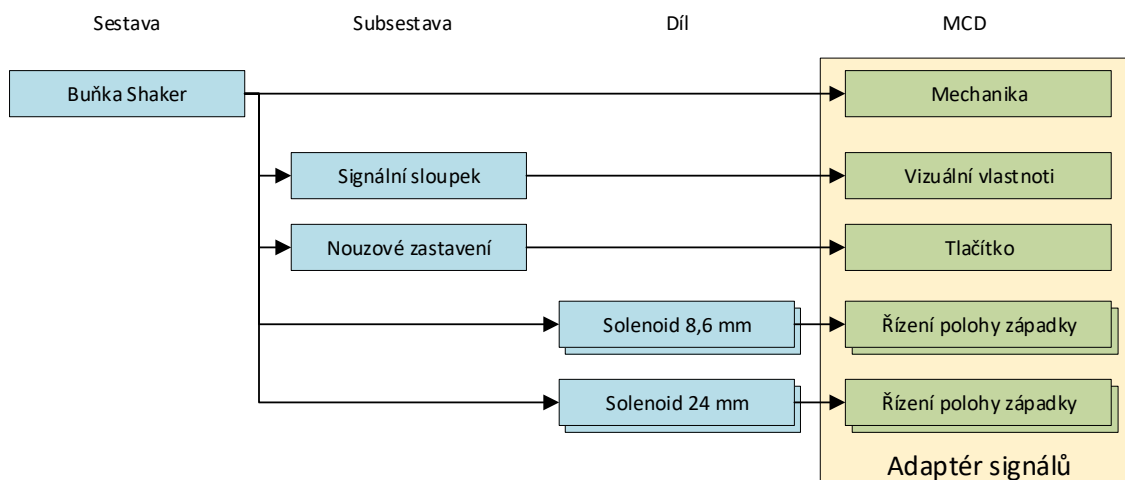
Obr. 21 Stromová struktura projektu

Postup práce v MCD odpovídá řazení stuktury projektu. Nejprve je nutné definovat základní fyziku těles a jejich kolize v *Basic Physics*. Druhým krokem je rozhýbání těles ve stanovených mezích spojů a vazeb, které se provádí v *Joints and Constraints*. Třetí část *Materials* je dedikována pouze pro materiály kolize. Ve čtvrté části *Couplers* jsou definovány spojky, převody a vačky, proto se v české verzi NX nazývá tato část „Spojky“. *Sensors and Actuators* sdružuje všechny prvky, které působí silou na fyzikální model (pohony), nebo v opačném případě model působí na ně (senzory). V části *Runtime Behaviors* je možné programovat logické nebo matematické bloky. Tyto logické bloky navázat s pohony nebo senzory v části *Signals*. A úplným finálem práce je připojení externích signálů na vnitřní signály modelu v části *Signal Connection*.

3.2.1.1 Vlastníci

Buňky jsou normalizovány nejen v rozměru a připojení na síť, ale i v prvcích rychlé interakce. Tyto prvky jsou standardizovány a využívají se ve všech buňkách. Bylo by neefektivní, vlastnosti vytvářet znovu pro každou buňku zvlášť. Modul MCD umožňuje vlastnosti přidělit sub sestavám nebo dílům. Pokud je v projektu více stejných dílů s definovanými vlastnostmi, MCD je očíslované zobrazí v hlavní sestavě. Z hlavní sestavy je nelze ovlivňovat, ale jsou dosažitelné pro mapování signálů. Změna vlastníka se provede pomocí příkazu  *Change Owner Component*.

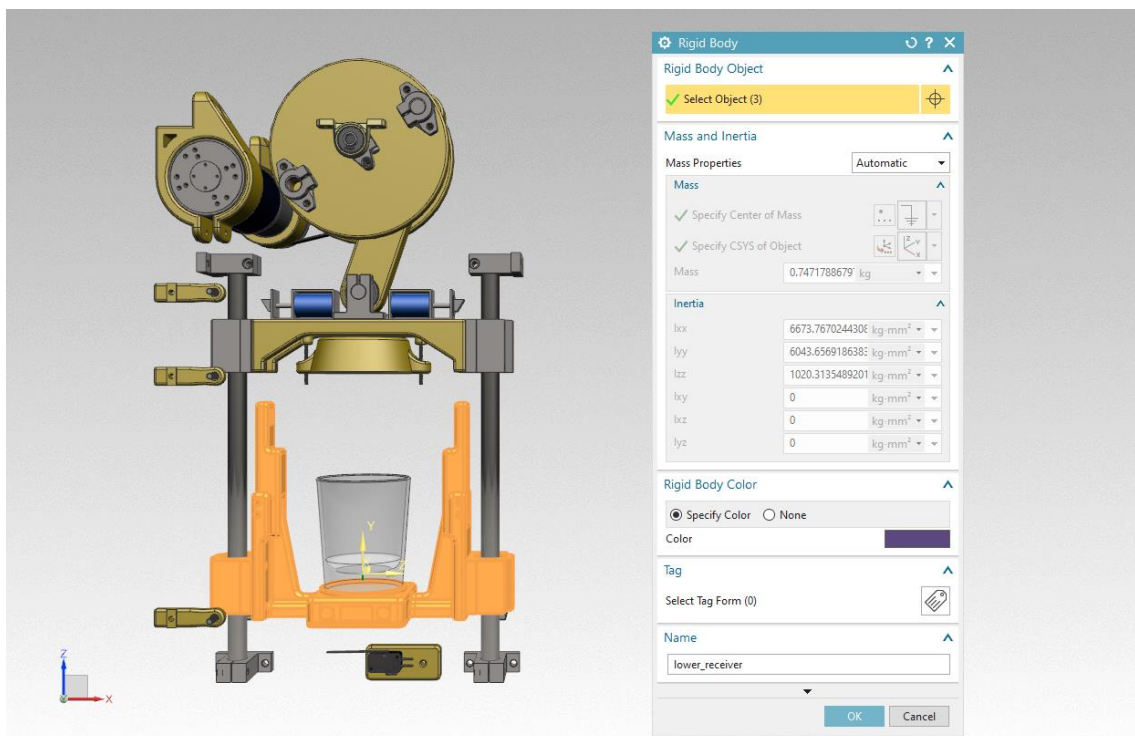
Sestava buňky shaker využívá dva normalizované komponenty: signální sloupek a tlačítko nouzového zastavení. Proto je jejich MCD profil přesunut do subsestav. Dále buňka používá vícenásobně použité díly, které mají komplexnější funkce. Jedná se o solenoidové západky. Proto je díl solenoidu vlastníkem svých funkcí. Ostatní MCD závislosti jsou vlastněny celou sestavou buňky shaker. V okénku *Physic navigator* je sloupec *Owner Component* (Komponenta vlastníka). V této kolonce můžeme najít, který díl, popřípadě sestava, je vlastníkem. U MCD vlastností, které vlastní pracovní sestava, je políčko prázdné.



Obr. 22 Organizace vlastníků

3.2.2 Nedeformovatelná tělesa

Nedeformovatelná tělesa jsou základním kamenem modulu MCD. Na nedeformovatelná tělesa se váží všechny kinematické i ostatní vlastnosti modelu. Jsou definovány uživatelem nebo se mohou generovat automaticky při tvorbě kinematické vazby. Na každé takto definované těleso působí gravitační zrychlení. Po spuštění simulace těleso padá volným pádem, není-li jinak ukotveno. V dialogovém okně funkce *Rigid Body* (Nedeformovatelná tělesa) je možné uživatelsky specifikovat hmotnost tělesa, lokaci těžiště a momenty setrvačnosti v osách, viz obrázek Obr. 23. Při realizaci byly tyto parametry určeny automaticky, jelikož hmotnosti těles jsou ovlivňovány nastaveným materiálem jednotlivých dílů sestavy. V první příloze existuje tabulka nedeformovatelných těles pro hlavní mechanismus buňky viz Tab. 8. Tato tělesa jsou vlastněna hlavní sestavou buňky.




Obr. 23 Nedeformovatelné těleso

Důležitým parametrem pro nedeformovatelná tělesa je jejich hmotnost a těžiště. Nastavení hmotnosti lze provést automaticky nebo manuálně. Automatický výpočet vychází z nastavené hustoty a objemu dílu. Manuální nastavení nám dovoluje využít fyzického měření hmotnosti. Nastavení materiálů je detailněji popsáno v kapitole 3.2.4.

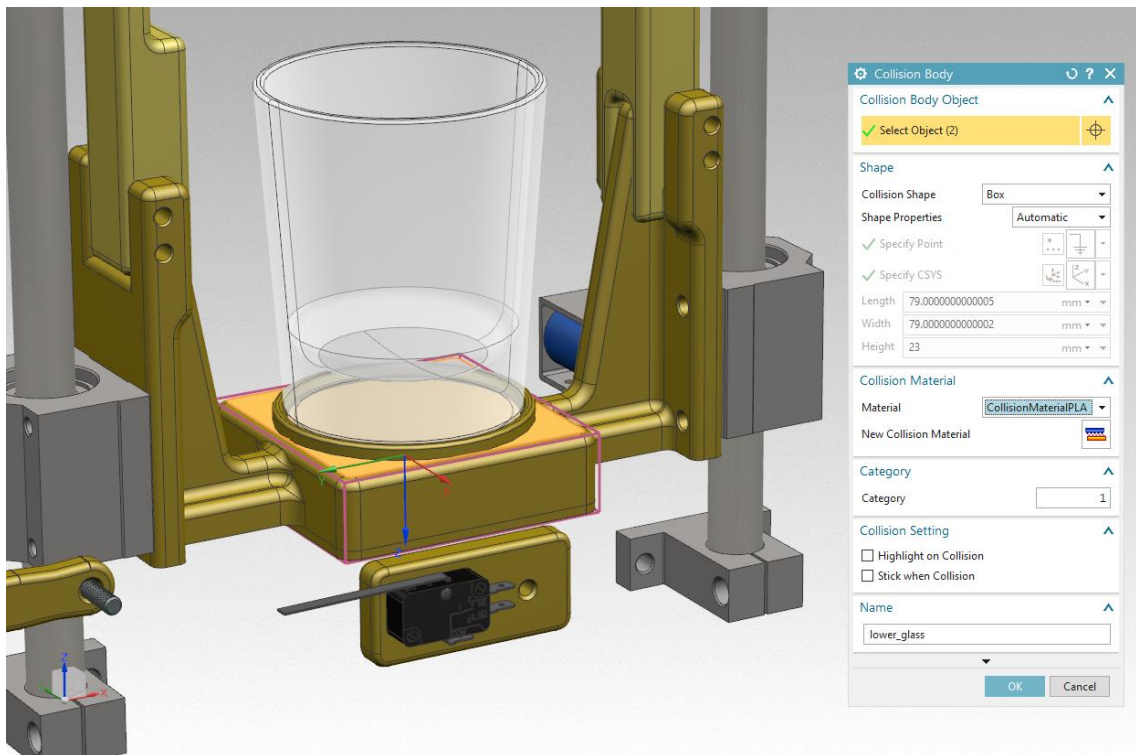
Hmotnosti je vhodné určit alespoň přibližně už v počátku návrhu. Všechny mechanické části stroje musí být poháněny, pokud mají vykonávat práci. Je pravděpodobné, že aktuátory definované podle reálného zařízení nebudou dostatečně silné.

3.2.3 Kolidující tělesa

Kolize lze simulovat pouze mezi kolidujícími tělesy. Ostatní tělesa jsou bezkolizní a s kolizními nijak neinteragují. Pokud kolizní těleso je součástí některého z nedeformovatelných těles, je tato vlastnost nedeformovatelnému tělesu organizačně podřízena. Při definici kolizních modelů se používají geometrické náhrady, takzvaných geometrických primitiv. Funkce *Collision Body*  umožňuje na výběr ze sedmi aproximací.

- 1) *Box* (Kvádr)
- 2) *Sphere* (Koule)
- 3) *Cylinder* (Válec)
- 4) *Capsule* (Válcový objekt se sférickým dnem)
- 5) *Convex* (Konvexní mnohostěny)
- 6) *Multi Convex* (Více konvexních mnohostěnu)
- 7) *Mesh* (Obecné těleso aproximované sítí)

Poslední tři aproximace jsou z pohledu výpočetní náročnosti nejsložitější. Proto je důležité, aby takto aproximovaná tělesa měla co nejmenší počet polygonů. Při překročení tří set aproximačních polygonů uživatele MCD upozorní. V takovém případě už není zaručena plynulost simulace. Vždy je potřeba brát ohled na co nejjednodušší řešení.




Obr. 24 Kolidující těleso

Pokud jsou kolidující tělesa příliš tenká, například při použití sítě jen na stěnu nedeformovatelného tělesa, není zaručeno správné chování ve smyslu fyziky srážky. Tělesa mohou v takovém důsledku po překročení malé síly penetrovat kolizní překážku.

V nastavení *Collision Body* je možné dále specifikovat materiál kolize viz kapitola 3.2.4 nebo kategorii kolize. *Category* je parametr, kterým lze kolize separovat. Této vlastnosti se dá využít u kolizních senzorů, které jsou senzitivní jen na určité materiály, viz kapitola 3.2.6.

3.2.4 Materiály simulace

Materiály použité v simulaci lze modifikovat ve dvou rovinách. První úroveň nastavení vychází z CAD modelu a tudíž je specifikována nejlépe konstruktérem. Jedná se o nastavení mimo modul MCD, ale na simulaci má značný vliv. Po nastavení materiálu dílu sestavy, ovlivníme automatický výpočet hmotnosti nedeformovatelných těles v MCD. Nastavení se provádí v *Menu – Tools – Materials – Assign Materials*. Můžeme použít knihovnické materiály nebo vytvořit vlastní. Nastavení parametru hustoty je dostačující pro tuto aplikaci.

Druhá rovina nastavení materiálů už souvisí přímo s modulem MCD. Jedná se o nastavení mechanických vlastností kolizních těles. V dialogu funkce *Collision Material*  (Materiál Kolize) jsou čtyři základní modifikovatelné parametry.

- Dynamické tření (*Dynamic Friction*)
- Statické tření (*Static Friction*)
- Valivý odpor (*Rolling Friction*)
- Koeficient pružnosti (*Restitution*)

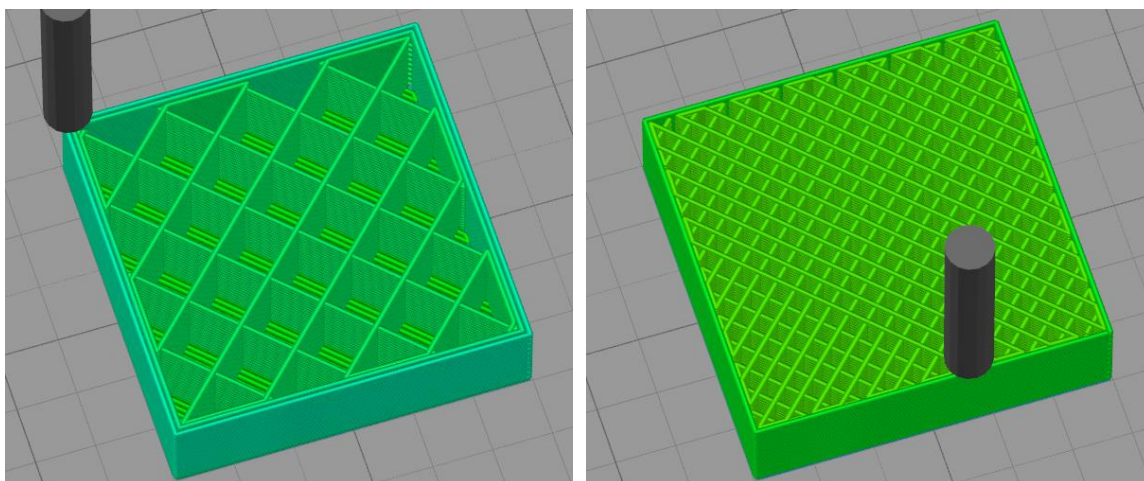
Významnou roli hraje nastavení při použití dopravníků (*Transport Surface*) nebo kluzných ploch. Koeficient restituce nebo pružnosti nám udává pružnost srážky. Pohybuje se v rozmezí 0 až 1, kde nulová restituce je pro tuhá tělesa a jedna pro tělesa elastická. Všechny kolize v modelu jsou definovány jako nepružné.

3.2.4.1 Hmotnost tištěných dílů

Tato kapitola se krátce zabývá 3D tiskem. Není to hlavní částí práce s MCD, ale je nutné brát v potaz některé jeho vlastnosti. V projektu Barman hraje 3D tisk hlavní roli. Ať už jako nástroj rychlého prototypování nebo jako levné nahrazení obráběných dílů. Je otázkou, zda by tento projekt byl realizován bez takzvaných „domácích“ 3D tiskáren (FDM/FFF). [18]

Hlavní roli při simulování dílů, vyrobených technologií FDM, hraje celková hmotnost výsledného tělesa. Zásadní vliv na ni má parametr INFILL, tj. procentuální hodnota výplně vnitřního objemu tělesa bez stěn. Stěny takového tělesa také ovlivňují celkovou hmotnost. Důležitý je jejich počet a tloušťka. Tento problém nelze zcela

odstranit ani v případě, že nastavíme *infill* na 100 %. Těleso není v celém objemu zcela homogenní. V plném tisku jsou mikrotrhliny, které nelze nijak odstranit.



Obr. 25 Určení hmotnosti 3D tištěných těles

Na Obr. 25 je vidět těleso v softwaru, který vytváří řídicí program pro 3D tiskárny (*Slicer*). Vlevo je krychle o straně 30x30x30 mm v průběhu tisku. *Infill* této krychle je 10 % a počet vnějších perimetrů je dva. Vpravo je ta samá krychle, ale *infill* je nastaven na 30 % a tiskne se pouze jeden vnější perimetr. Jednou z funkcí takového programu je i výpočet hmotnosti spotřebovaného materiálu, pokud v nastavení uvedeme hustotu materiálu.

Jednodušším a v tomto případě i vhodnějším řešením je tedy díly zvážit. Nejlépe po postprocesingu, jako je třeba broušení nebo vrtání. Jelikož nebyla možnost jednotlivé díly zvážit, hmotnosti byly zvoleny heuristickým přístupem. Zpětně už nebylo možné dohledat tiskové parametry a každý díl má jinou hodnotu parametru *infill*, která se mění od 50 % do 100 %. Proto bylo přibližně zvoleno 70 %. Tento parametr se promítl do snížení standardní hustoty použitého materiálu. Při optimalizaci modelu bude nutné stroj rozebrat a díly zvážit.

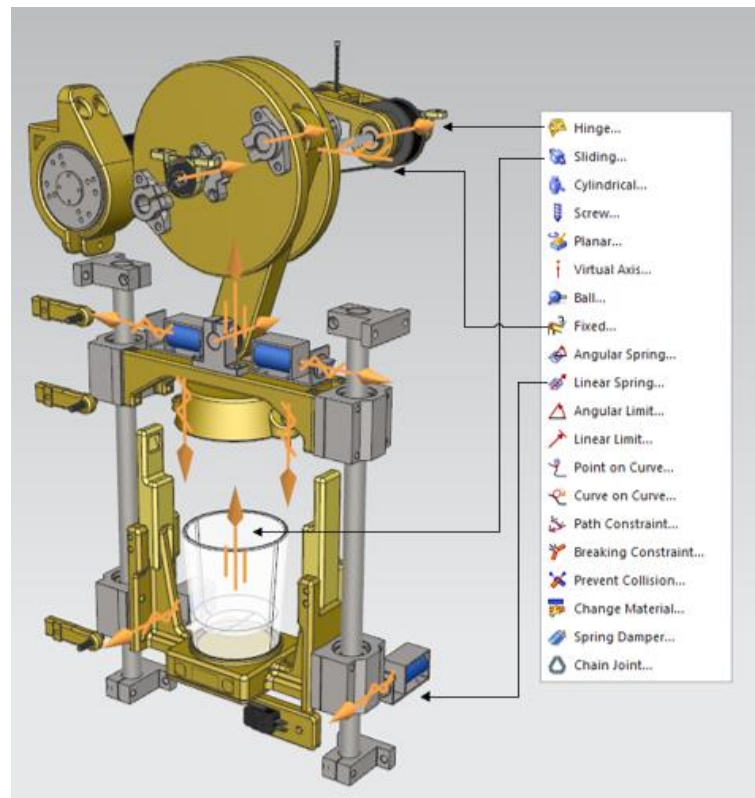
Technologie FDM má více úskalí, například silně anizotropní mechanické vlastnosti takto vytvořených dílů. V této práci, ale nebylo účelem zjišťovat pevnostní a únavové koeficienty jednotlivých dílů. Tento obor je výhradou strojní fakulty. Bylo by zajímavé v budoucnu popsat tyto parametry a pomocí prostředí NX simulovat i možné destrukce.

3.2.5 Spoje a vazby

Tvorba spojů se skládá ze dvou nedeformovatelných těles. Základem je těleso, na které se spoj aplikuje (*Attachment*). Druhé těleso tvoří základnu (*Base*). Těleso základny není povinné. Pokud není definované, spoj je zakotven vůči hlavnímu souřadnicovému systému sestavy. Pro *Fixed Joint* (Pevný spoj) výběr nedeformovatelných těles dostačuje, ale pro další spoje je nutné určit další specifikace.

Osa nebo také vektor definuje například vazbu *Sliding Joint* (Lineární osa pohybu). Nejjednodušší vytvoření vektoru je výběrem z hlavního souřadnicového systému. Tato možnost je výhodná z pohledu kompatibility vazby. Například po výměně CAD modelu, na kterém je vazba definována. Všechny vektory ale není možné definovat tímto způsobem. Dalším způsobem definice je odvození z geometrie modelu. Prostředí obsahuje funkce pro chytré definice vektorů. Využít lze definice ze dvou bodů, tečny křivky, osy nebo normály k ploše.

Pro další vazby, jako například *Hinge Joint* (Pant) nebo *Ball Joint* (Kloub), bude nutné definovat i bod ukotvení. Body můžeme definovat souřadnicově nebo využít chytrých funkcí. Metody odvození od: středu kružnice, vrcholu tělesa, konce hrany, tečny nebo středu křivky.



Obr. 26 Mechanické spoje a vazby

3.2.6 Senzory a aktuátory

Collision Sensor 🧲 (Senzor kolize) je v nové verzi NX vybaven novými schopnostmi. Nejvýznamnější vlastností je možnost nastavení spouštění. To umožňuje implementace „chytřejších“ senzorů s vnitřní logikou. Možnosti spouštění jsou:

- *Trigger*

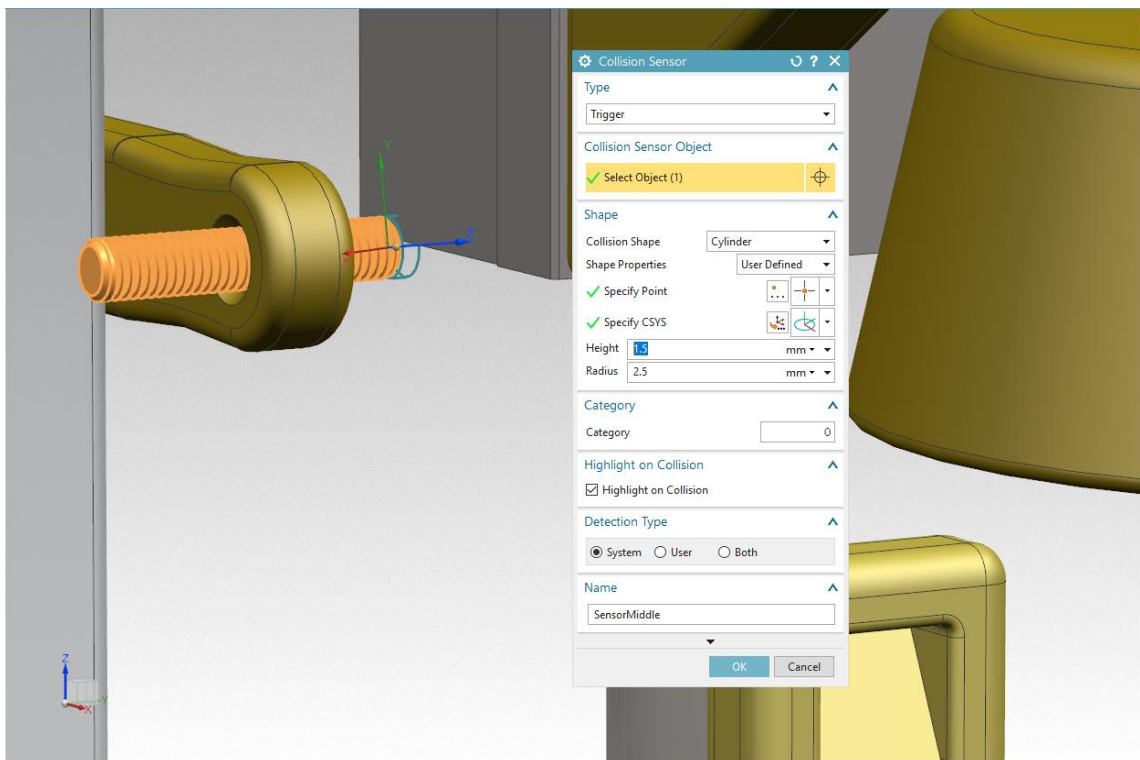
Standardní spouštění na aktivní úroveň. Vychází z NX v12, po migraci projektu jsou všechny snímače typu Trigger

- *Switcher*

Náběžná hrana události změny stav výstupní hodnoty snímače. Snímač kopíruje vlastnosti klopného obvodu typu T. V kombinaci s logickým blokem lze nakonfigurovat senzor s vnitřní logikou například anti-windup, kumulativní čítací senzory a další.


Další novou vlastností funkce *Collision Sensor* je iniciovat kolizní senzor uživatelem, kliknutím na daný senzor. V části nastavení *Detection Type* je možné separovat možnosti inicializace senzoru. Pokud má kolizní senzor definovanou interakci s uživatelem (*User/Both*), je možné ho ovládat například z VR prostředí. Výhodou je také to, že si snadno dokážeme prověřit správnost namapovaných signálů.

Snímač kolize nemusí být jen kontaktní. Kolizní model snímače lze rozšířit mimo tělo senzoru. V našem případě se jedná o bezkontaktní snímání polohy. V závislosti na principu senzoru můžeme aproximovat jeho zorné pole. Na výběr je cylindrická, sférická, kubická aproximace kolize. Paprsek laserových snímačů můžeme aproximovat přímkou.

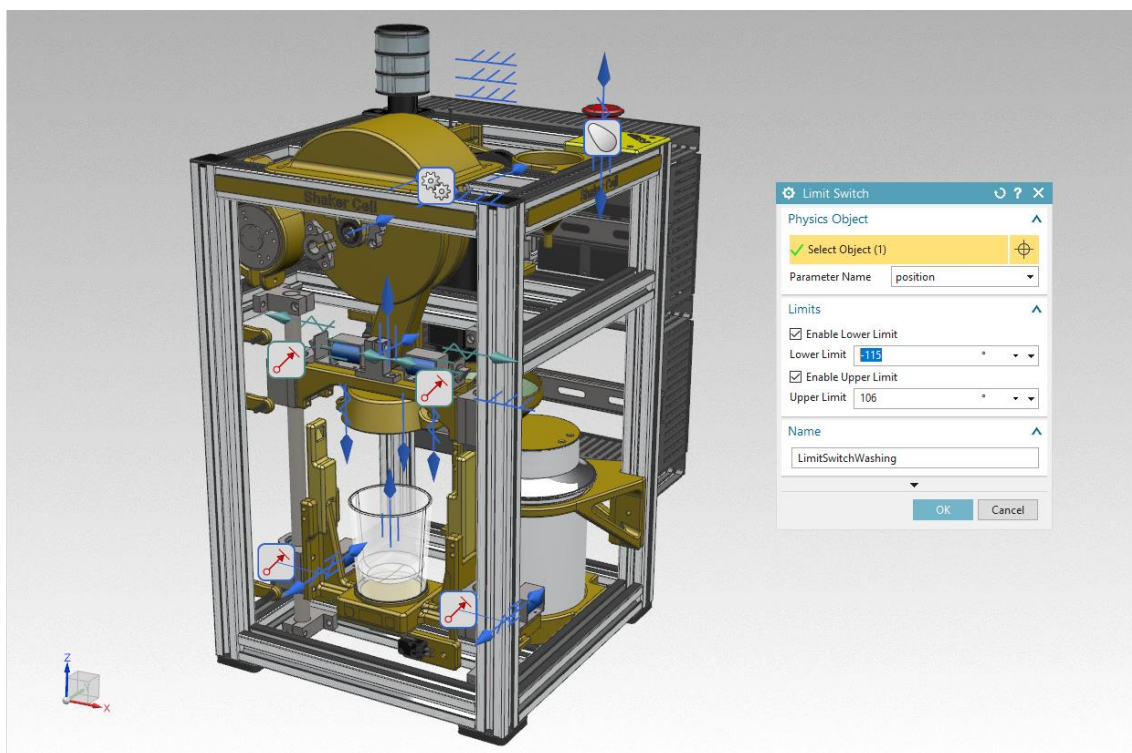


Obr. 27 Senzor kolize (bezkontaktní snímač polohy)





V nastavení *Shape* vybereme tvar kolize na *Cylinder* (válcová). Pokud máme kontaktní senzor *Shape Properties* ponecháme na *Automatic*. V našem případě je použit bezkontaktní senzor, kolizní model je nutné „vytáhnout“ z těla senzoru, a proto jsou vlastnosti definovány jako *User Defined*. Celá inicializace se skládá z koordinačního souřadného systému (*Specify CSYS*) a středu tělesa (*Specify Point*).

Limit switch  (Limitní spínač) je další funkcí vhodnou pro definování senzoru polohy. Výhodou oproti koliznímu snímači je to, že není potřeba funkci aplikovat na nedeformovatelné těleso. Funkce může být definována na jakémkoliv parametru simulace, jako například poloha, rychlost, zrychlení nebo síla. Tyto parametry jsou součástí vazeb, převodovek nebo vaček.

Spínače můžeme definovat na jakémkoliv trajektorie stroje. V obrázku Obr. 28 je vidět „*Limit Switch Washing*“, to je ve fyzické buňce koncový mikrospínač detekující polohu mycí misky. Pokud je signál od spínače aktivní, mycí miska je v poloze pod víčkem sklenice. Buňka je ve stavu mycího cyklu. V dialogu funkce se nastaví hystereze sepnutí.



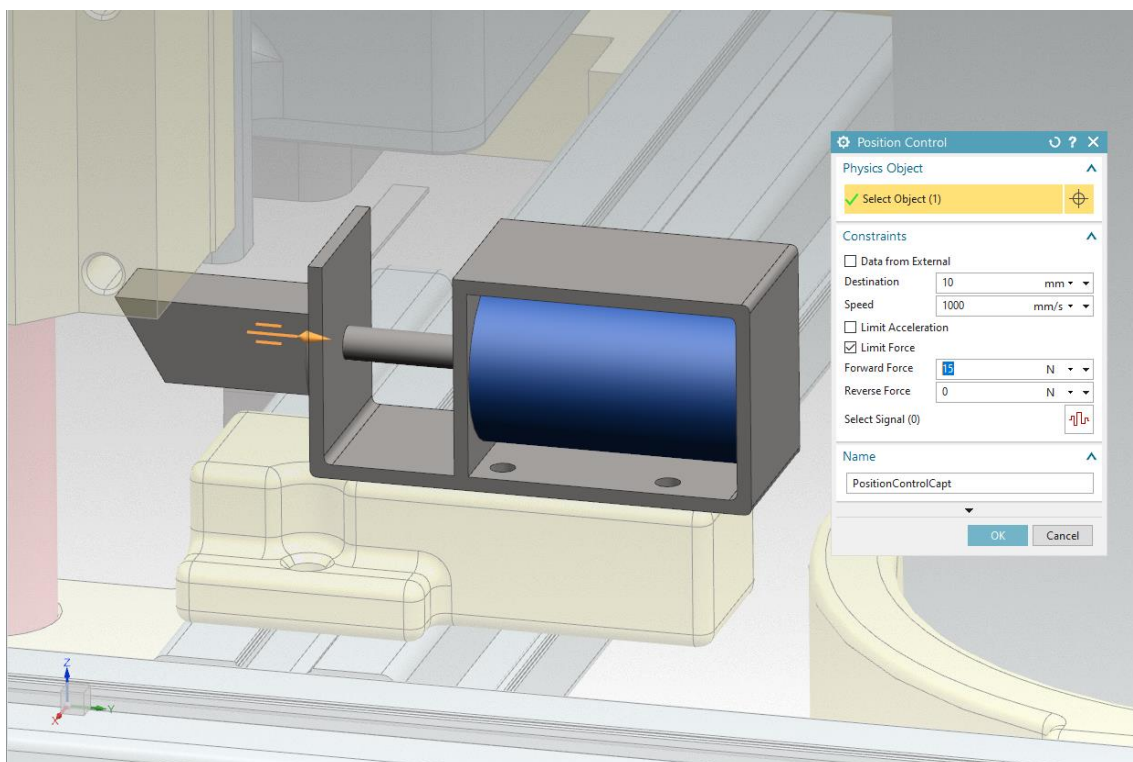
Obr. 28 Limitní senzory

Pro snímání spojitych veličin má MCD celou řadu dalších funkcí, například *Distance Sensor* , *Position Sensor* , *Velocity Sensor* , *Accelerometer*  nebo obecný *Generic Sensor*, který lze aplikovat na jakýkoliv parametr simulace.

Můžeme tak měřit i veličiny, které fyzické zařízení neměří. Využití takových dat je v optimalizaci procesů. Simulace veličin, které fyzicky neměříme, je vhodné v případě,

kdy by bylo měření nepřiměřeně drahé nebo fyzicky nerealizovatelné. V ideálním případě by bylo možné dosáhnout i bezsnímačového řízení.

Virtuální akční členy jsou v modelu použité jak rotační, hlavní motor a mycí motor, tak i lineární elektromagnetický zámek (solenoid). Ovládání západky solenoidu je realizováno pomocí funkce **Position Control** (Řízení pozice). Výrobce elektromagnetického zámku udává vzdálenost vtažení západky 10 mm a maximální sílu elektromagnetu 15 N. Západka nedokáže dosáhnout omezující rychlosti na dráze 10 mm. Funkce řízení pozice se váže na lineární spoj *Sliding Joint*, v případě zámku nebo jinou vazbu s jedním stupněm volnosti, například rotační vazbu *Hinge Joint*.



Obr. 29 Řízení pozice solenoidu

Motor i s převodkou má výrobcem udané maximální otáčky 152,7 ot/min a kroučící moment 5,83 Nm. Tyto parametry jsou definovány v dialogovém okně funkce **Speed Control** (Řízení rychlosti). Důležité je zmínit, že tato funkce se musí aplikovat na fyzikální vazbu s jedním stupněm volnosti, v tomto případě rotační vazbu *Hinge Joint* pro první řemenici („gear_in“). Poté je nutné vytvořit řízení rychlosti pro opačný směr a nulovou rychlost, který celou sestavu brzdí, je-li motor ve vypnutém stavu.


Tímto docílíme motorizaci náhonu řemenového převodu. Pro převod na druhou kladku řemenice („gear_out“) je využita převodovka. Funkce převodovky sice není součástí *Sensors and Actuators*, ale je organizována v sekci „Spojky“ *Couplers*.


Poznámka:

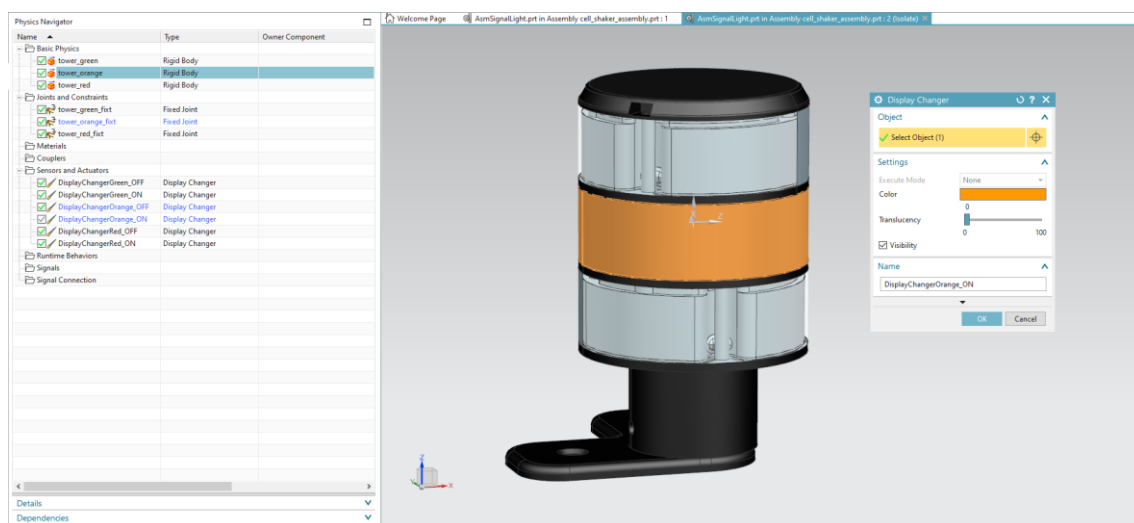
Jazykový překlad názvů funkcí je oficiální z NX verze 12 a originální anglické popisy jsou z migrace na verzi 1903. Některé automatické překlady nejsou zcela výstižné. Například importu signálu dovoluje možnost „KROK7“, jedná se o produkt značky Siemens STEP7.

Ve funkci převodovky *Gear* pouze zvolíme primární rotační vazbu *Master*, to je ta, na které je definováno řízení rychlosti od motoru. Druhou z vazeb je *Hinge Joint* pro druhé kolo převodu (*gear_out*) *Slave Axis Joint*. Parametry převodu *Master Multiple* a *Slave Multiple* jsou nastaveny podle reálných hodnot převodového poměru.

$$1: 1,25 \quad \rightarrow \quad \text{Master Multiple: Slave Multiple} \quad (1)$$

V nové verzi MCD je možné za tímto účelem použít novou funkci *Pulleys and Belts* , která simuluje celé řemenové převody. Tato funkce by byla vhodná i pro tuto buňku, ale neměla by zpomalovat simulaci. Funkce převodovky chod simulace nijak výrazně nezpomalí. Pokud by byla simulace náročná na výpočet, nastalo by přesimulování modelu, je zde cesta pro zjednodušení simulace. Řízení rychlosti by se v takovém případě mohlo přepočítat na motor i s převodovkou.

Signální sloupek je realizován pomocí funkce *Display Changer* . Funkce při aktivaci umožňuje změnit barvu dílu. Proces změny barvy je však nevratný stejnou funkcí. Pro přepínání barev je nutné použít dvě funkce, jednu pro „rozsvícení“ a druhou pro „zhasnutí“. Logika externího signálu je naprogramována v adaptéru signálů. Jelikož je signální sloupek použit u všech buněk, jsou tyto funkce realizovány v sestavě signálního sloupku. Díky tomu funkce v ostatních buňkách nemusíme znovu vytvářet.



Obr. 30 Komponenta vlastníka a změna vzhledu

3.2.7 Signály a mapování

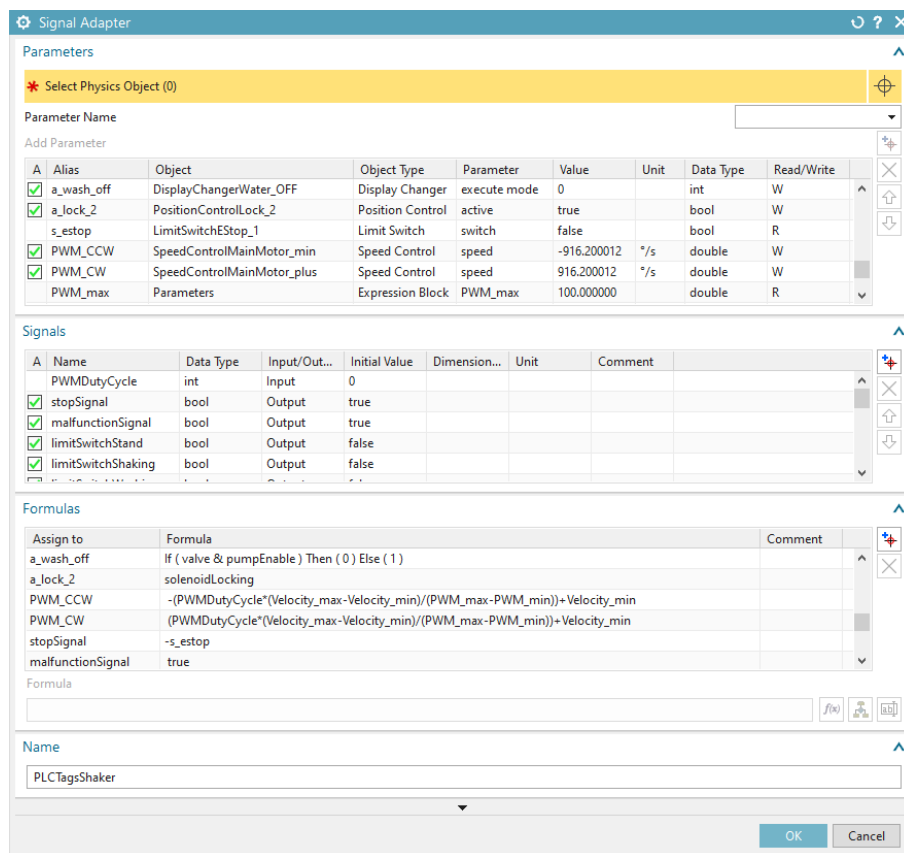
Mapované signály jsou převzaty z bakalářské práce pana Karmiše [17]. Export signálů proběhl z prostředí TIA Portal, ve kterém jsou vytvořeny všechny řídicí programy pro projekt Barman. Z „Tag Table IOs“ signály převedeme do formátu XLSX pomocí příkazu *export*. Tento formát lze do MCD nahrát trojím způsobem. Prvním způsobem je využití exportovaných dat pro import do adaptéru signálu (*Import Signal Adapter*). Druhý způsob je přímý import signálů do *Import Signals*. Posledním využitím je ve funkci *SymbolTable*, kde je možné importovat tabulku symbolů, neboli jmen signálů. Všechny tyto funkce pracují se soubory XLSX. Importovat lze i z jiných zdrojů, jako například SIMIT nebo STEP7.

Tab. 6 Signály buňky Shaker

Jméno signálu	Type	Adresa	Funkce
stopSignal	Bool	%I0.0	Stop tlačítko
malfunctionSignal	Bool	%I0.1	Výpadek napájení buňky
limitSwitchStand	Bool	%I0.2	Koncový spínač
limitSwitchShaking	Bool	%I0.3	Mycí talíř v poloze, kdy není aktivní
limitSwitchWashing	Bool	%I0.4	Mycí talíř v poloze mytí
inductiveSwitchTop	Bool	%I0.5	Snímač polohy horní úvrati
inductiveSwitchMiddle	Bool	%I0.6	Snímač střední polohy
inductiveSwitchBottom	Bool	%I0.7	Snímač polohy dolní úvrati
beaconGreen	Bool	%Q0.0	Signalizace zelená
beaconOrange	Bool	%Q0.1	Signalizace oranžová
beaconRed	Bool	%Q0.2	Signalizace červená
mainMotorEnable	Bool	%Q0.3	Spuštění hlavního motoru
washMotorEnable	Bool	%Q0.4	Spuštění motoru mycího talíře
motorDirection	Bool	%Q0.5	Směr motorů
pumpEnable	Bool	%Q0.6	Spuštění mycí pumpy
solenoidLocking	Bool	%Q0.7	Zámky na spojení horní a dolní části
solenoidStand	Bool	%Q1.0	Zámky pro usazení odkládací části
valve	Bool	%Q1.1	Ventil mycí sekce
PWMDutyCycle	Int	%QW1008	Hodnota střídání hlavního motoru [%]

Vlastní signály je možné vytvořit v dialogu funkce *Signal*. MCD definuje tři základní datové typy interních signálů: binární signál (*bool*), 32 bitový celočíselný integer (*int*) a 64 bitový signál s plavoucí desetinnou čárkou (*double*). Signál s desetinnou tečkou MCD považuje za inženýrskou hodnotu, a proto data prezentuje i s jednotkou SI.

Model využívá signálů importovaných přes *Signal Adapter* (Adaptér signálu). Ten tvoří rozhraní pro parametry simulace a vnitřní signály modelu. Vnitřními signály jsou přes logické nebo matematické funkce propojeny s parametry, aby se s nimi mohlo nadále pracovat při běhu simulace. Řízení buňky je zprostředkováno binárními signály, pouze hlavní motor, řízený pomocí PWM modulace, bylo nutné upravit. Procentuální hodnota střídá se předává modelu pomocí proměnné „*PWMDutyCykle*“, která je typu integer. Postup práce v adaptéru signálu je dělen do tří tabulek v dialogovém okně. Do první tabulky přidáváme runtime parametry simulace. Všechny fyzikální prvky simulace obsahují parametry, které jsou dostupné uživateli pro tento účel. Přidáme je označením fyzikálního objektu a výběrem jeho jména. Tlačítkem *Add* vložíme parametr do první tabulky adaptéru. V druhé tabulce máme námi importované signály. Ty můžeme stejným způsobem přidávat nebo editovat. Třetí a nejdůležitější tabulka obsahuje formule (*Formulas*), neboli logické nebo matematické funkce, které propojí prvky z prvních dvou tabulek. Formulí vytvoříme tak, že zaklikneme check box u výstupního signálu (*Output*) nebo parametru pro zápis (*Write*). Následně v příkazovém řádku *Formula* naprogramujeme požadovanou funkci z ostatních prvků v adaptéru. Rychlý náhled do adaptéru signálů je vidět na obrázku Obr. 31. Celý adaptér pro buňku shaker je umístěn v přílohách na straně 70 a 71.



Obr. 31 Náhled adaptéru signálu


Logika řízení motoru je rozdělena do dvou rovin. Logické funkce ovlivňují aktivaci jednotlivých řízení rychlosti. V této části jsou vytvořeny tři, jedna pro kladný směr otáčení, druhá pro záporný směr otáčení a třetí brzdný s nulovou rychlostí. Pokud je signál „*Direction*“ v logické 1, motor se po aktivaci „*mainMotorEnable*“ otáčí po směru hodinových ručiček. Druhá rovina řízení je přidělena změně parametru rychlosti. Je nutné přepočítat procentní hodnoty PWM modulační na rychlost ve stupních za sekundu. Jedná se o standardní lineární převod do inženýrských jednotek, viz obrázek Obr. 31. Parametry výpočtu jsou předány pomocí *Function Blok*. Zde jsou uloženy a kopírovány na výstupy bloku, aby je bylo možné měnit za chodu simulace nebo snáze upravovat.

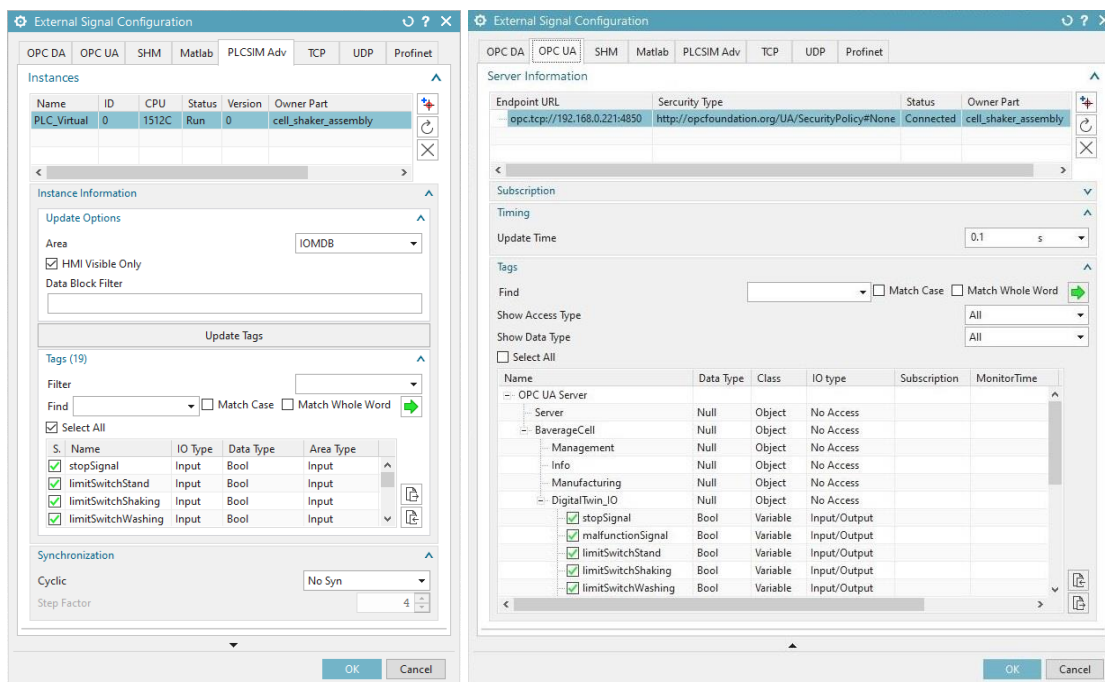
- *PWM_max*
- *PWM_min*
- *Velocity_max*
- *Velocity_min*

Signály „*malfunctionSignal*“ a „*stopSignal*“ jsou zdroji globálního chybového hlášení v řídicím programu buňky. Oba jsou z důvodu bezpečnosti aktivní v logické nule. Proto musí být inicializovány při startu simulace ve stavu logická jedna (*true*). Nastavení probíhá v adaptéru signálu, tabulka *Signals*, sloupec *Initial Value*. Díky tomu řídicí program nedetekuje restart simulace. První ze zmíněných signálů „*malfunctionSignal*“ ve fyzickém zařízení detekuje výpadek napájení silového obvodu. V našem simulovaném případě je proto nevyužit a nachází se trvale v logické úrovni jedna.

3.2.8 Externí signály

V prvním kroku konfigurace externího propojení budeme muset zprovoznit komunikační protistranu. V našem případě se jedná o virtuální PLC nebo OPC server. Pokud je PLC aktivní, můžeme se pustit do propojení s MCD.

Druhým krokem bude otevření dialogového okna příkazu *External Signal Configuration* ^{ENT} (Externí konfigurace signálu). Ze záložek vybereme možnost *PLCSIM Adv* a tlačítkem *Add*  otevřeme okno *Add PLCSIM Adv Instance*. V okně vidíme jméno aktivního zařízení a jeho status. Po stisku tlačítka OK se vrátíme zpět do okna externí konfigurace signálu viz levá část obrázku Obr. 32.



Obr. 32 Externí konfigurace signálu

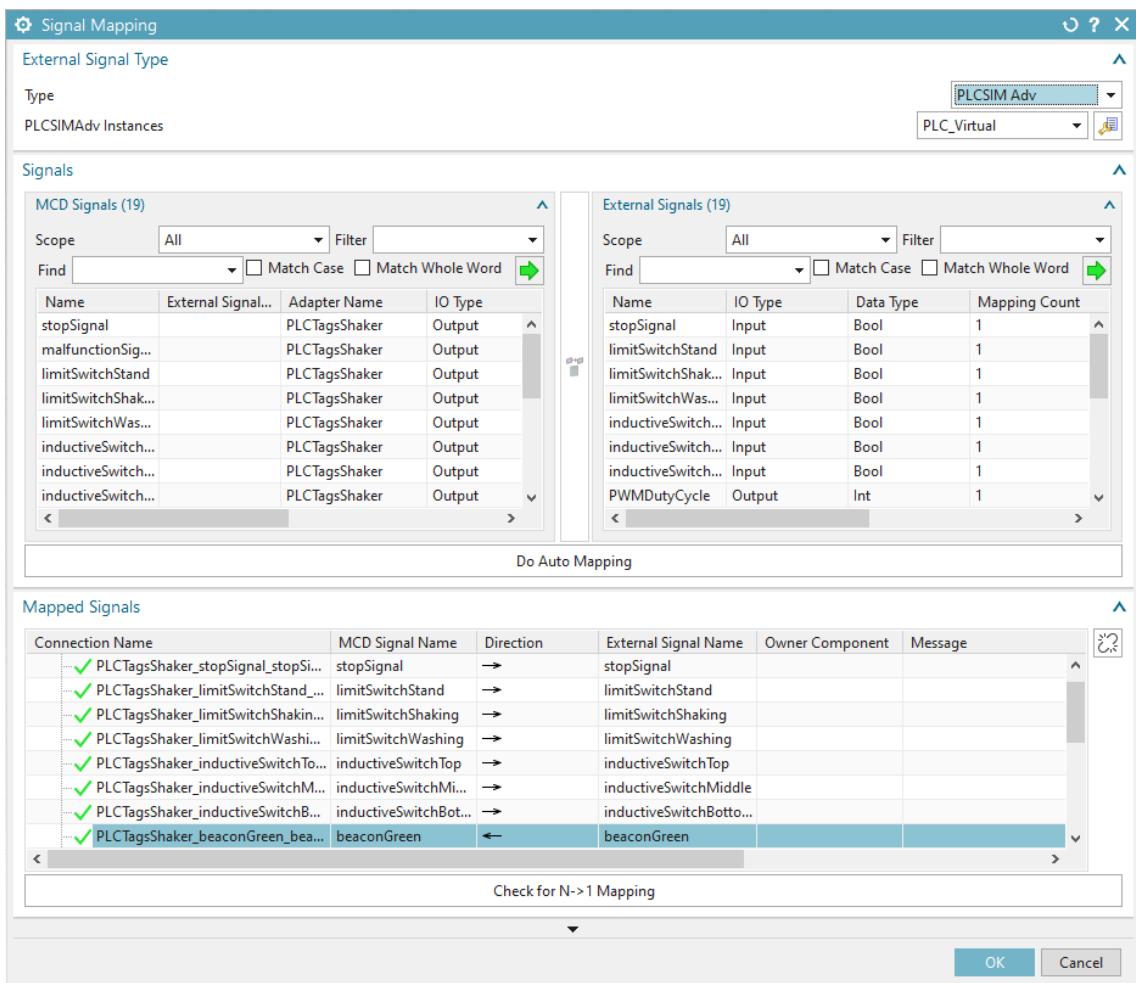
Při hledání signálů je možné specifikovat parametry výběrových filtrů. V části *Update Options* vybíráme, v jakém paměťovém prostoru PLC budeme signály hledat. Je možné vyhledávat ze vstupů a výstupů (IO), paměťových merkerů (M) nebo datových bloků (DB). V části *Tags* označíme pouze ty signály, které chceme mapovat. Pro přesné řízení simulace je možné nastavit synchronizaci.

Pokud budeme mapovat signály z OPC UA serveru, je nejprve nutné spustit aplikaci NX jako správce. Druhou podmínkou pro úspěšné propojení je přidání výjimky z firewallu. Při vytváření externí konfigurace signálu v záložce OPC UA přidáme další zdroj, jak tomu bylo v předchozím případě. Otevře se dialogové okno, do kterého zadáme adresu koncového bodu serveru. Z možností zabezpečení vybereme první možnost bez zabezpečení (*None*). MCD v aktuální verzi 1903 nepodporuje jiné možnosti zabezpečení. Signály pro mapování vybereme ze struktury tagů serveru.

Externí signály mohou být ve zdroji definovány těmito datovými typy: Bool (1 bit), Byte (1 byte), Word (2 byte), Int (2 byte), DWord (4 byte), DInt (4 byte), Real (4 byte) a LReal (8 byte). Signály se automaticky převádí na vyšší přesnost a je jim přidělena jednotka podle interního signálu v MCD.

3.2.9 Mapování signálů

Posledním krokem je mapování signálů. Po úspěšné konfiguraci příkazem *Signal Mapping* (Mapování Signálů) otevřeme okno viz obrázek Obr. 33. Zde vybereme typ externího signálu, který byl nakonfigurován v předchozím kroku. Díky tomu se v části *External Signals* zobrazí žádané signály. Pokud jsou jména signálů v MCD shodná se jmény externích signálů tlačítkem *Do Auto Mapping*, je automaticky spojíme. Ručně je možné vazbu vytvořit tlačítkem *Map Signal* a zrušit vazbu pomocí tlačítka *Break*. Pro kontrolu můžeme ověřit, zda jsou signály správně namapovány ve smyslu jeden signal zdroje do jednoho signálu příjemce. K tomu slouží funkční tlačítko *Check for N → 1 Mapping*.



Obr. 33 Mapování signálů

3.3 Řízení virtuálního modelu

Tato kapitola je shrnutím toho, jaký hardware řídicí jednotky lze využít pro řízení modelu v MCD. Samozřejmě ne všechny možnosti jsou použitelné ve stávajícím projektu Barman. Buňky stroje jsou řízeny pomocí PLC od firmy Siemens, produktové řady S7-1200. Přesné označení použitého typu je 1214C DC/DC/DC. Nevýhodou této řídicí jednotky je firmware, který není schopen komunikovat přes standard OPC. Tento problém je tedy vyřešen externě mimo řídicí jednotku. Druhou nevýhodou je nemožnost emulovat PLC v nástroji PLCSIM Advanced. Většina projektů, prezentujících možnosti nástroje MCD, využívá PLC řady 1500 právě z důvodu optimalizace propojení s emulovanou řídicí jednotkou. Pro potřeby testování virtuálního modelu buňky je použito PLC dostupné v laboratoři jak fyzicky, tak virtuálně 1512C-1 PN.

Tab. 7 Možnosti propojení s PLC

typ	Profinet	OPC UA	OPC DA	PLCSIM Adv.	TCP	UDP
S7-1512C-1PN	ano	ano	ano	ano	ano	ano
S7-1214 C	ano	ne	ne	ne	ano	ano

S propojením pomocí OPC DA nebo Profinet nebylo uvažováno. Prvním důvodem je fakt, že OPC server pro výrobní systém projektu Barman je vypracováván ve standardu OPC UA. OPC server je tvořen na platformě Linux, která není pro DA kompatibilní. A posledním důvodem je i odklon od tohoto standardu v praxi. Metoda propojení skrze standard Profinet je ze strany MCD omezena, viz kapitola teoretické části 2.4.1.

3.3.1 Řídicí program SiL

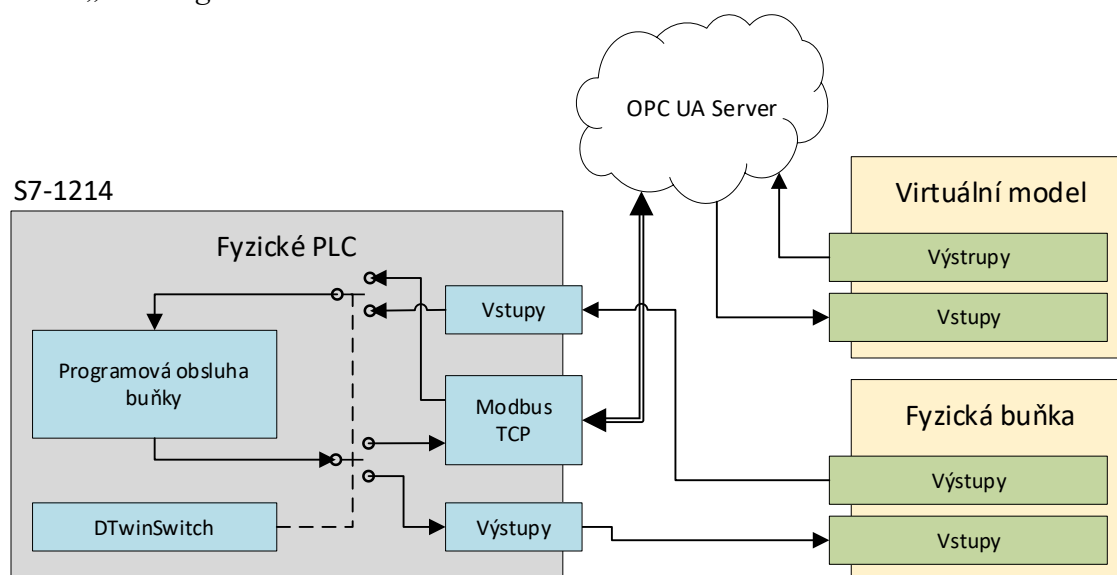
PLC použité k řízení fyzické buňky shaker (S7-1214) je vybaveno modulem s rychlými výstupy pro PWM modulaci. Program musel být upraven pro použití automatu (S7-1512). Tato řídicí jednotka neobsahuje tento modul, a proto ani programové vybavení pro jeho obsluhu. V PLC S7-1214 řeší obsluhu modulátoru systémový datový blok „CTRL_PWM_DB“. Aby bylo možné řídicí program migrovat do nového PLC, musel být tento blok i jeho volání odstraněno.

Následně bylo upraveno volání tohoto bloku ve funkčním bloku „ShakingMotorFC“. Úprava se týkala předávání parametrů do dříve zmíněného datového bloku. Tato část kódu byla odstraněna a nahradí ji pouze přiřazení stavu „ShakingMotor_Struct.State.Enabled“ na výstup %Q0.3. PWM výstupu v simulaci není možné obsloužit z časového hlediska, viz kapitola 3.2.7. Další úpravy v kódu nejsou žádoucí z důvodu toho, aby logika fyzického řízení byla totožná s virtuální.

3.3.2 Řídicí program HiL

Druhý z programů je pro původní řídicí jednotku S7-1214. Program integruje virtuální model buňky do konceptu virtuálního dvojčete metodou HiL. OPC server je propojen s řídicími jednotkami buněk skrze rozhraní Modbus TCP a vytváří prostředníka mezi řídicí jednotkou a modelem. Pro tento účel je v řídicí jednotce vytvořen datový blok „*BaverageCell*“, který organizuje komunikační signály pro výrobní buňky. Do zmíněného datového bloku byla přidána struktura signálů „*DigitalTwin_IO*“ pro virtuální vstupy a výstupy MCD. Důležité je rozdělit vstupy a výstupy v adresném prostoru, aby nedocházelo k přepisu paměti. Komunikační buffer je dvoubajtový, a proto musí být rezervovaný adresný prostor vstupů a výstupů jeho násobkem. Pokud by se na jednom bajtu vyskytovaly vstupy i výstupy, OPC server by při aktualizaci vstupů mohl přepsat i námi změněné výstupy.

Na obrázku Obr. 34 je názorně vidět propojení reálných vstupů s virtuálními. Toto propojení je doplněné o přepínač. Přepínač je realizován v PLC kódu. Každý vstup nebo výstup je vybaven přepínačem řízeným pomocí jednoho paměťového merker „*DTwinSwitch*“. Pokud je přepínač v logické nule, jsou „odstrženy“ fyzické vstupy a výstupy PLC a připojují se s virtuálními vstupy a výstupy v komunikačním datovém bloku „*BaverageCell*“.



Obr. 34 Koncept HiL

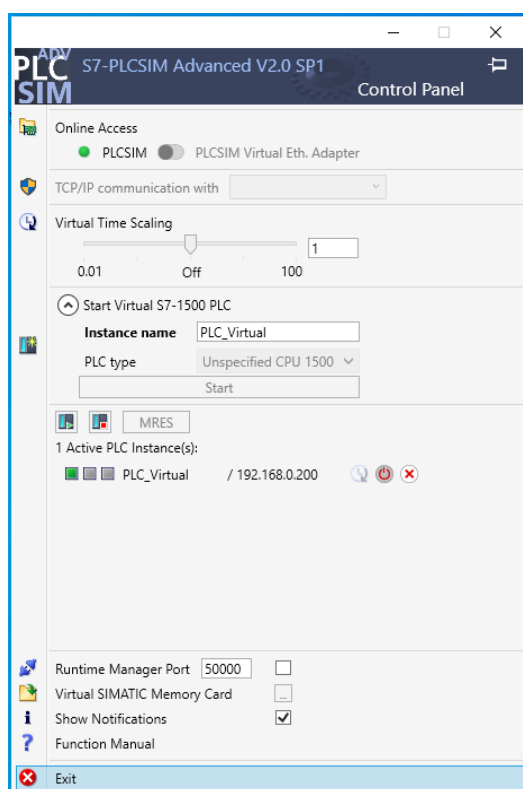
Jak je vidět z obrázku Obr. 34, řídicí jednotka obsluhuje vždy jen model nebo fyzickou buňku, nikoliv obě zároveň. Bylo by možné dosáhnout jistého „paralelismu“ i v koncepci HiL. Pokud uvážíme to, že řídicí program vytěžuje PLC z méně než padesáti procent, jak je tomu ve většině případů, je možné program pro buňku vykonávat dvakrát. Popřípadě by se mohla použít další fyzická řídicí jednotka. Pokud je řídicí program napsán dle standardů pro dávkové řízení (S88), není problém použít stejný funkční blok pro jednotku (buňku) a virtuální jednotku (model). Virtuální vstupy a výstupy by v takovém případě byly namapovány stejným způsobem, jen by je nebylo

nutné přepínat. Přepínač by nebyl potřebný, ale z bezpečnostního hlediska by bylo vhodné program doplnit o odpínač fyzických výstupů. Ten by mohl být realizován stejně jako tomu je u přepínače. Pokud by toto řešení nebylo žádoucí, je nutné při ladění s virtuálním modelem aktivovat tlačítko nouzového zastavení na fyzické buňce.

3.3.3 Virtualizace PLC

PLCSIM Advanced je simulační nástroj, který je schopen vytvořit virtuální obraz PLC. Software dovoluje simulovat pouze řídicí jednotku typu S7-1500 a jednotku distribuovaných vstupů ET 200SP. Na rozdíl od standardního PLCSIM, které simuluje programové vybavení, je simulace obohacena o emulaci fyzického komunikačního rozhraní. Tento nástroj je tudíž jedinou možností pro simulaci komunikace mezi více prvky v průmyslové síti, například profinet nebo v omezené míře OPC. Výhodou PLCSIM Advanced je i kompatibilita s nástroji pro virtualizaci.

Pro tuto práci je použit program S7-PLCSIM Advanced V2.0 SP1. Při zapnutí je otevřen kontrolní panel viz obrázek Obr. 35. Pro navázání virtuálního PLC s modelem v MCD metodou sdílené paměti stačí spustit PLCSIM mód. Vytvoření virtuálního síťového adaptéru je možné v případě, že by do PLC měl mít přístup nadřazený systém. Pro ověření simulovaného modelu tato možnost není potřeba. I přesto že není vytvořen síťový adaptér, je nutné definovat IP adresu zařízení. Tato adresa musí být shodná s adresou PLC v rámci prostředí TIA Portal. Tlačítkem start spustíme virtuální PLC a nahrajeme řídicí program, který byl popsán v kapitole 3.3.1.



Obr. 35 PLCSIM Advanced

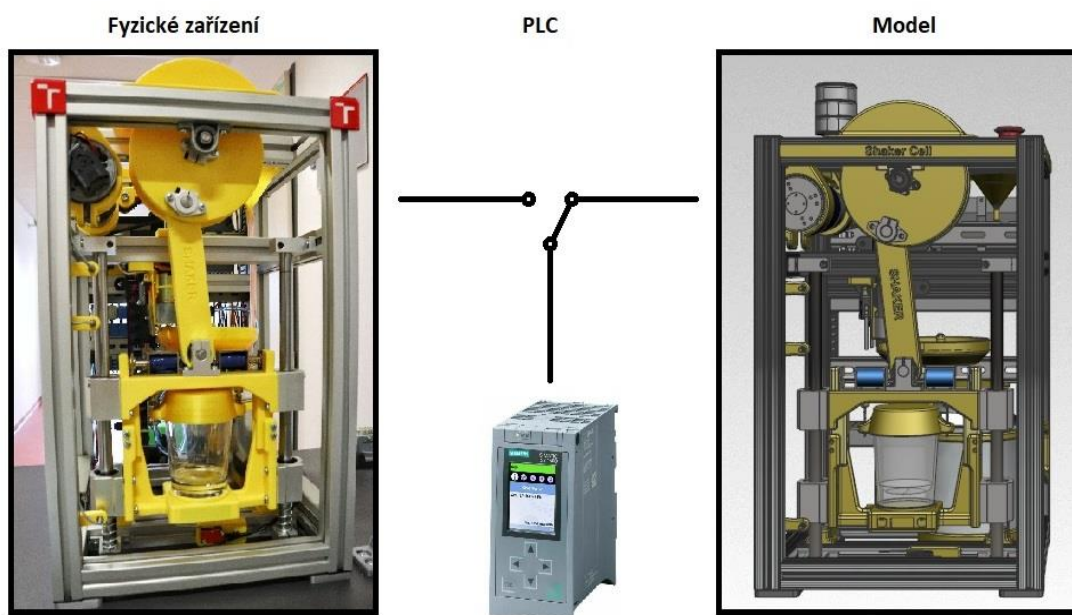
Další možností virtualizace PLC je skrze program SIMIT. Ten je schopen propojení i s jednodušší variantou PLCSIM. Nelze sice simulovat komunikaci přes OPC nebo Profinet, ale je možné simulovat programové vybavení i PLC řady S7-1200.

3.4 Ověření funkce

Ověření funkce virtuálního modelu probíhalo metodou *Software in the Loop*. Model buňky byl propojen s virtuální řídicí jednotkou v PLCSIM Advanced.

Nejprve probíhaly testy spojení jednotlivých vstupů a výstupů. Takzvané Loop Test's, které ověřily funkčnost spojení, logiku virtuálních snímačů a akčních členů. Při těchto testech byly provedeny úpravy v logické části modelu. Bylo nutné upravit brzdný moment mechanismu. Ten simuluje tření mechanismu a odpor převodové kaskády. Po ověření signálů byl proveden cyklus míchání v autonomním režimu, jako je tomu při fyzické výrobě.

V další fázi proběhl test propojení se serverem OPC UA. Tato část testů se zaměřovala na ověření konceptu integrace digitálního dvojčete do systému stroje. Stroj používá řídicí systém řady S7-1200, proto nebylo možné operovat s metodou SiL, ale digitální dvojče se integruje pomocí metody *Hardware in the Loop*.



Obr. 36 Schéma konceptu

Ověření spojení s OPC serverem proběhlo s využitím fyzického PLC S7-1512C-1PN. Ve výsledném konceptu se ale tento typ PLC nenachází, proto dalším krokem byl test s externím serverem. Ve spolupráci s panem Bc. Magáthem, který OPC server vytvářel pro svoji diplomovou práci, se podařilo docílit stabilní komunikace fyzického PLC s virtuálním modelem v MCD.

3.5 Řešené problémy

Tato kapitola se zabývá problémy, které nastaly při tvorbě virtuálního zprovoznění jak v programu TPS, tak i v modulu MCD. Problémy jsou výčtem činností, na které je dobré brát ohled při návrhu.

3.5.1 Problémy s TPS

Zásadním problémem při integraci virtuálního modelu manipulátoru byla neexistence virtuálního kontroléru pro tento typ robota. Řešení se hledalo v externím řízení robotických operací. Díky rozdílům v přístupu TPS a fyzického robot kontroléru by integrace musela probíhat přes prostředí třetí strany. Přenositelnost robotického programu by byla neproveditelná. Aplikace by nesplňovala požadavky na virtualizaci, ale jednalo by se pouze o efektní 3D vizualizaci.

Robot je v konceptu stroje ovládán z PLC pomocí zpráv po sériovém rozhraní RS-232. Zprávy se skládají z příkazů, které odkazují na body uložené v paměti robota. Robot poté sám vypočítá cestu, kterou pro dosažení cíle použije. Robotické operace v TPS definují celou cestu. Proto by bylo nutné vytvořit všechny kombinace cest mezi buňkami. Toto řešení bylo prohlášeno za nevhodné z důvodu zpětné kompatibility mezi fyzickým strojem a simulací. Částečné zjednodušení bylo hledáno v přidání uzlových bodů. To by zmenšilo počet potřebných operací, ale nevyřešilo rozhraní simulace.

Při realizaci externího propojení přes OPC UA nebyly signály dosažitelné, i když měly shodný název. Ukázalo se, že signály v TPS musí mít název v uvozovkách, aby je prostředí dokázalo namapovat. Po této úpravě bylo spojení úspěšně navázáno s fyzickou jednotkou PLC S7-1512.

3.5.2 Problémy s MCD

Při ověřování kolizních modelů nastával předem neočekávaný jev. Objekty ignorovaly své kolize i při malých působících silách. Důvodem byly tenké kolizní modely u obou součástí. Nechtěný jev se projevoval nejen na kolizích součástí, ale i na funkci kolizních senzorů. Pokud volíme u kolizních součástí aproximační metody sítě, je vhodné je nedefinovat pouze na jedné ploše, ale přidat do této aproximace třetí rozměr (objem). Problém nastává, pokud budeme definovat vnitřní prostor (koš, krabičku). Poté je potřeba využít právě síťové, konvexní nebo multi-konvexní aproximace. Druhá z dvojice součástí musí být definována na celý objem.

Tření je obecně definováno pomocí materiálů simulace. Vhodné je této vlastnosti využít pro transportní plochy a dopravníky. Tření mechanismu buňky (lineární ložiska, odpor převodovky) bylo vyřešeno pomocí řízení rychlosti pohonu a jeho brzdného momentu. Směr otáčení motoru je řízen signálem „*Direction*“. Logika signálu přepíná mezi řízením rychlosti s kladným a záporným smyslem otáčení. Pomocí signálu „*Eneable*“ je spuštěna funkce s nulovou rychlostí. Díky tomu je mechanismus schopen zastavit, pokud na něj motor aktivně nepůsobí.

S prvními pokusy o realizaci tření mechanismu vyvstalo další omezení MCD. Zmíněný pokus simuloval i tření kluzné tyče a ložiska standardně pomocí materiálu simulace. Tyč lze aproximovat primitivním tělesem válcového tvaru, ovšem aproximace válcové dutiny ložiska je možné realizovat pouze konvexním nebo síťovým modelem. Při aproximaci dochází k tomu, že kruh je zjednodušen polygonem s konečným počtem hran. To zmenší kolizní průměr součásti a tolerance mezi součástmi zanikají. V důsledku kolize tak dochází k penetraci součásti a k nepředvídatelným stavům po spuštění simulace.

Stejný problém byl zaznamenán u víčka sklenice a sklenice samotné. Zde je nutné docílit vhodné tolerance mezi sklenicí a osazením víčka. Problém vyřešila až úprava průměru kolizního tělesa. Nejde o úpravu CAD modelu sklenice, ale výběr příslušných kolizních ploch. Dalším opatřením je i rozlišení této kolize pomocí jiného kolizního ID. Tím docílíme separaci od ostatních kolizí, které by mohly ovlivnit další funkce nebo budoucí rozšíření.

Dalším problémem simulace je PWM řízení hlavního motoru. Z důvodu vysoké frekvence PWM je v simulaci použit digitální vstup, který udává procento střídavy modulace. Tato hodnota ovlivňuje parametr rychlosti funkce *Speed Control* hlavního motoru. Kód v PLC nebylo nutné upravovat, jelikož výstup s touto hodnotou byl již použit. Reálný dvoustavový vstup PWM pro řízení motoru, je možné použít jako logickou funkci „*Eneable*“ nebo ho v simulaci nevyužít.

Přístup k vizuální stránce signalizačního majáku nebyl obdobný jako v ostatních řešeních. Funkce *Display Change* nelze spustit pomocí parametru „*Active*“ jako většinu funkcí, ale parametrem „*Execute mode*“. *Execute mode* je parametr typu integer, ale změna se provede při zápisu hodnoty větší než jedna. Důležité je vědět, že funkce se po aktivaci vykoná a není zvrtná. Nelze barvu změnit a stejnou funkcí ji poté odstranit. Je nutné definovat dvě funkce, jednu na změnu aktivní barvy (červená, zelená, oranžová), druhou na změnu na neutrální barvu (zhasnuto). Funkce je namapována na signál s použitím jednoduché podmínky v adaptéru.

Dalším řešeným problémem bylo externí propojení OPC UA. Prostředí NX musí být spuštěné v režimu správce a nesmí být blokováno firewallem operačního systému. Identifikovat takový problém lze pomocí externího OPC klienta, který neměl problém se spojit se serverem. Oproti tomu funkce pro mapování externích signálů v MCD nedokázala nalézt žádný aktivní server.

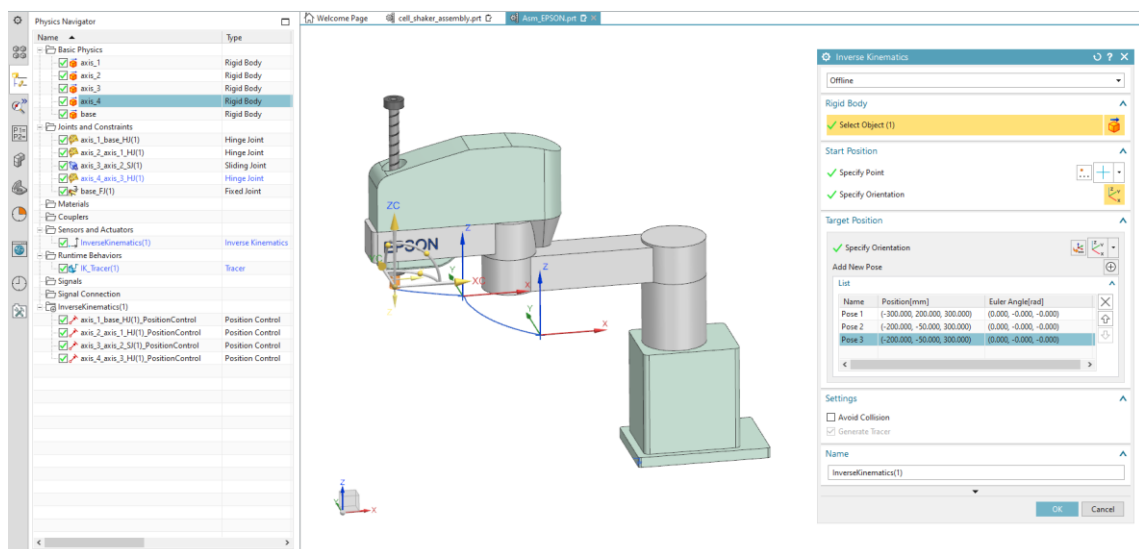
3.6 Inverzní kinematika

Tato kapitola nesouvisí s prací na buňce Shaker, ale má za cíl poukázat na možnost řešení částečné virtualizace robotického manipulátoru v MCD. V nové verzi NX 1903, která je nyní dostupná i pro potřeby projektu Barman, přišla firma Siemens s implementací funkce inverzní kinematiky. Tato a další nové funkce značně zasahují do působnosti systému TPS.

Funkce Inverse Kinematics ↕ (Inverzní kinematika) je zařazena do skupiny aktuátorů. Je aplikována na nedeformovatelné těleso, v ukázkovém příkladu je výpočet odvozen od poslední osy robota. Pokud by model obsahoval gripper, bylo by nutné přidat k této ose ještě statické části gripperu (tělo gripperu). Specifický bod by byl v takovém případě mezi chapadly v místě uchycení sklenice. Dialog funkce nabízí ze dvou módů výpočtu kinematiky *Online* a *Offline*.

Online mód, jak už plyne z názvu, provádí výpočet kinematiky za chodu simulace. Pozice kam se má kinematická sestava přestavit, je předána z runtime signálů, například ze sekvenčního grafu nebo externího signálu.

Offline kinematika je vypočtena ještě před startem simulace. Nejprve je nutné definovat body trajektorie. Funkce inverzní kinematiky vytvoří automaticky novou složku v projektovém stromu a naplní ji řízením polohy pro všechny osy robota, viz obrázek Obr. 37.



Obr. 37 Inverzní kinematika v MCD

4. ZÁVĚR

Cílem této práce bylo vytvořit virtuální dvojče části stroje Barman a navrhnout způsob jeho integrace do celkového konceptu.

Teoretická část práce se snaží seznámit čtenáře s pojmem Průmysl 4.0, jehož neodmyslitelnou částí je i virtualizace. Hlavní část pojednává o základních virtuálních metodách *Hardware in Loop* a *Software in the Loop*. Práce se zabývá i průzkumem na trhu virtuálních řešení. Důraz je kladen na popis obou softwarů uvedených v zadání práce. Jaké jsou jejich aplikační možnosti, výhody a nevýhody. V neposlední řadě práce seznamuje se školním projektem TestBed Barman, jehož část byla prakticky virtualizována.

V rámci zadání, nejprve semestrální práce, se počítalo s realizací v prostředí TPS. Pro demonstraci byla vybrána transportní jednotka, robotický SCARA manipulátor EPSON. I přes počáteční obtíže bylo nalezeno možné řešení. Nicméně řešení by bylo složité a nepřineslo by tížený výsledek v podobě digitální kopie robota. Přístup v řízení fyzického a simulovaného robota v TPS je odlišný a realizace by neodstranila problém s přenositelností programů mezi strojem a jeho virtuálním obrazem. Z tohoto důvodu se od dokončení ustoupilo.

V závislosti s tímto rozhodnutím se naskytl příležitost pracovat v modulu MCD, který je integrován v prostředí NX. Tato skutečnost byla natolik inovující, že došlo k modifikaci zadání. Jako demonstrační úloha byla vybrána výrobní buňka Shaker. Ověření fyzikálních vlastností modelu bylo realizováno metodou SiL s použitím virtuálního kontroléru emulovaného v PLCSIM Advanced. Jelikož tento nástroj nedokáže virtualizovat PLC použité pro řízení fyzické buňky, musel se koncept ve výsledku zaměřit na druhou metodu HiL. K tomu byla vytvořena metodika a její implementace je popsána v této práci. Virtuální model využívá propojení s OPC serverem, který je součástí projektu. Výstupem této práce je tedy funkční virtuální model buňky Shaker, použitelný jako digitální dvojče. Pro potřebu ověření funkce modelu byl upraven řídicí program fyzické buňky. Vznikly dva programy pro obě virtuální metody. Důležité je zmínit, že úpravy byly pouze ve struktuře, ne v logice programu. Simulační model reagoval na zásahy zmíněného programu správně. Využití této práce je především v budoucí optimalizaci řídicích algoritmů, výuce obsluhy a ověřování principů Průmyslu 4.0.

Na základě znalostí nabytých při realizaci šejkru byla rozpracována i buňka skladu sklenic, která se potýkala s problémem přesimulování. Gripper této buňky je složen z velkého počtu mechanických článků. Výsledná simulace je sice efektní po vizuální stránce, ne však efektivní pro výpočet. Dvouosý manipulátor je ve své vertikální ose poháněn korkovým motorem, který přes trapézový šroub pohybuje ramenem. Daleko přesněji bude nutné nastavit hmotnost ramene. Nejlépe celou buňku rozebrat a díly zvážít. Přichycení rozpínacího gripperu se sklenicí nejspíše nepůjde simulovat fyzikálně, protože i přes nastavení vysokého tření kolize, sklenice sklouzává. Řešení se zdá být ideální v programovém ovládní funkce *Hinge Joitn*. Dalším problémem je

stohování sklenic, kde nastává stejný problém, který byl zaznamenán i v reálné buňce. Sklenice se při složení zaseknou.

V závěrečné fázi práce proběhla nečekaná migrace projektu v MCD do nové verze. Původní verze NX 12 byla nahrazena verzí 1903. Kromě lehkých problémů s dokumentací to bylo správné rozhodnutí, protože nová verze NX přináší další možnosti. Vylepšené funkce, které jsou součástí simulace, jsou zadokumentovány v textu práce. Zajímavá pro projekt Barman je nově implementována funkce *Inverse Kinematics*, díky které by bylo možné docílit vizualizace robotického manipulátoru. Nová funkce dovoluje online i offline programování. Tím MCD smazalo velkou nevýhodu oproti systému TPS.

Literatura

- [1] PÁSEK, Jan. *Digitální transformace průmyslu* [online]. Brno, 2019 [cit. 2020-03-02]. Dostupné z:
https://www.vutbr.cz/www_base/priloha_fs.php?dpid=183201&skupina=dokument_priloha. Skripta. FEKT VUT.
- [2] INICIATIVA PRŮMYSL 4.0. In: *MPO* [online]. Praha: Ministerstvo Průmyslu a Obchodu, 2016 [cit. 2020-01-04]. Dostupné z:
<https://www.mpo.cz/assets/dokumenty/53723/64358/658713/priloha001.pdf>
- [3] Industry 4.0. In: *CAD Schroer* [online]. Cambridge: CAD Schroer, 2016 [cit. 2020-01-04]. Dostupné z: <https://www.cad-schroer.com/solutions/industry-4-0/>
- [4] PARK, Sang C., Chang Mok PARK, Gi-Nam WANG, Jongeun KWAK a Sungjoo YEO. PLCStudio: Simulation based PLC code verification. In: *2008 Winter Simulation Conference* [online]. IEEE, 2008, 2008, s. 222-228 [cit. 2020-04-13]. DOI: 10.1109/WSC.2008.4736071. ISBN 978-1-4244-2707-9. Dostupné z: <https://ieeexplore.ieee.org/document/4736071/>
- [5] *Informační zpravodaj AXIOM TEC* [online]. 2015, **2015**(16) [cit. 2020-03-08]. Dostupné z: <https://www.axiomtech.cz/wcd/zpravodaj/zpravodaj2015.pdf>
- [6] Virtuelle Inbetriebnahme mit NX11 Mechatronics Concept Designer: Presentation. In: *YouTube* [online]. Germany: AALE, 2017 [cit. 2020-04-15]. Dostupné z:
https://www.youtube.com/watch?v=g324gacUtlE&list=PLv6k0_48ZlBq6J4xdaxgOGjkopgWpeNCW
- [7] Komunikace přes rozhraní OPC UA. In: *Promotic* [online]. Ostrava: MICROSYS, spol. s r.o., 2014 [cit. 2020-01-04]. Dostupné z:
<https://www.promotic.eu/cz/pmdoc/Subsystems/Comm/OPC/OPCUA.htm>
- [8] HAMPL, L. Návrh a realizace robotických operací v systému Tecnomatix Process Simulate. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 81s. Vedoucí semestrální práce Ing. Václav Kaczmarczyk, Ph.D..
- [9] Tecnomatix. In: *SIEMENS ingenuity for life* [online]. Munich: Siemens Industry Software, 2019 [cit. 2020-01-04]. Dostupné z:
<https://www.plm.automation.siemens.com/global/en/products/tecnomatix/>
- [10] Process Simulate. In: *ONEPLM* [online]. Oxford: OnePLM Limited, 2017 [cit. 2020-01-04]. Dostupné z: <https://oneplm.com/tecnomatix/process-simulate/>
- [11] KACZMARCZYK, Václav, Ondřej BAŠTÁN, Zdeněk BRADÁČ a Jakub ARM. *An Industry 4.0 Testbed (Self-Acting Barman): Principles and Design*. IFAC-PapersOnLine [online]. 2018, 51(6), 263-270 [cit. 2020-03-05]. DOI: 10.1016/j.ifacol.2018.07.164. ISSN 24058963. Dostupné z:
<https://linkinghub.elsevier.com/retrieve/pii/S2405896318309108>

- [12] BUCHAL, D. Virtuální model stroje Tecnomatix Process Simulate. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 65 s. Vedoucí bakalářské práce Ing. Jakub Arm.
- [13] SÝKORA, Tomáš. Konstrukce a programové vybavení transportních entit pro testbed Průmysl 4.0. Brno, 2019, 79 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Václav Kaczmarczyk, Ph.D.
- [14] PROKOP, M. *Model třídící linky v systému Tecnomatix Process Simulate*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 75 s. Vedoucí diplomové práce Ing. Václav Kaczmarczyk, Ph.D..
- [15] LARSSON, ANDREAS a HENRIK NILSSON. *Visualize an event-based simulation model made in Process Simulate*. Sweden, 2011. Dostupné také z: <http://publications.lib.chalmers.se/records/fulltext/143057.pdf>. Master of Science Thesis. Department of Signals and Systems Division of Automatic Control, Automation and Mechatronics CHALMERS UNIVERSITY OF TECHNOLOGY.
- [16] VILLAGÓMEZ GUERRERO, Luis, Virgilio VÁSQUEZ LÓPEZ a Julián ECHEVERRY MEJÍA. Virtual Commissioning with Process Simulation (Tecnomatix). *Computer-Aided Design and Applications*. 2014, **11**(S), S11-S19. Dostupné také z: [http://www.cad-journal.net/files/vol_11/CAD_11\(S\)_2014_S11-S19.pdf](http://www.cad-journal.net/files/vol_11/CAD_11(S)_2014_S11-S19.pdf)
- [17] KARNIŠ, Radim. *Návrh, konstrukce a programové vybavení autonomní buňky "Shaker" pro testbed Průmyslu 4.0*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. Jan Pásek, CSc.
- [18] NOVOSÁD, Ivan. *Mechanické vlastnosti PLA dílů realizovaných 3D tiskem* [online]. Brno, 2018 [cit. 2020-02-04]. Dostupné z: <http://hdl.handle.net/11012/82099>. Bakalářská práce. Vysoké učení technické v Brně. Fakulta strojního inženýrství. Ústav konstruování. Vedoucí práce David Paloušek.
- [19] KALINA, Daniel. *Simulace výukových sestav strojů s využitím NX Mechatronics Concept Designer* [online]. Praha, 2018 [cit. 2020-04-13]. Dostupné z: <https://dspace.cvut.cz/handle/10467/80534>. Diplomová práce. ČVUT – Fakulta strojní. Vedoucí práce Ing. Lukáš Novotný, Ph.D.

Seznam příloh

Příloha 1 – Tabulka nedeformovatelných těles

Příloha 2 – Adaptér signálů – Signály

Příloha 3 – Adaptér signálů – Formule

Příloha 4 – Obsah přiloženého CD

Příloha 1 - Tabulka nedeformovatelných těles

Tab. 8 Nedeformovatelná tělesa

Nedeformovatelné těleso	Funkce
Sekce míchání	
upper_receiver	Horní část držáku sklenice je vybavena západkovým mechanismem pro spojení s dolní částí (lower_receiver)
lower_receiver	Dolní část držáku sklenice s odkládacím místem, jeho součástí jsou drážky pro spojení s horní částí
wheel	Klikový převod se skládá ze dvou planetových kol s excentrickým uložením kliky, kolo je hnáno výstupní řemenicí (gear_out)
crank	Klika
cup	Víčko sklenice je pružně spojeno s horním držákem (upper_receiver) a tvoří tenzi mezi sklenicí a spodním držákem (lower_receiver) při uzamčeném spojení
gear_in	Vstupní kolo řemenového převodu je poháněno hlavním motorem s převodovkou
gear_out	Výstupní kolo řemenového převodu přes hřídel pohání klikový mechanismus
Sekce čištění	
washcup	Výchylná miska s oplachovou tryskou
water	Vizualizace hladiny vody

Příloha 2 - Adaptér signálů – Signály

Signal Adapter

Parameters

* Select Physics Object (0)

Parameter Name

Add Parameter

A	Alias	Object	Object Type	Parameter	Value	Unit	Data Type	Read/Write
	s_middle	SensorMiddle	Collision Sensor	triggered	false		bool	R
	s_top	SensorTop	Collision Sensor	triggered	false		bool	R
	s_bottom	SensorBottom	Collision Sensor	triggered	false		bool	R
	s_stand	limitSwitchStand	Collision Sensor	triggered	false		bool	R
<input checked="" type="checkbox"/>	a_capt_1	PositionControlCapt_1	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_capt_2	PositionControlCapt_2	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_lock_1	PositionControlLock_1	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_lock_2	PositionControlLock_2	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_p	SpeedControlWashMotor_plus	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_m	SpeedControlWashMotor_min	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_z	SpeedControlWashMotor_zero	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_p	SpeedControlMainMotor_plus	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_m	SpeedControlMainMotor_min	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_z	SpeedControlMainMotor_zero	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_red_on	DisplayChangerRed_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_red_off	DisplayChangerRed_OFF_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_orange_on	DisplayChangerOrange_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_orange_off	DisplayChangerOrange_OFF_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_green_on	DisplayChangerGreen_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_green_off	DisplayChangerGreen_OFF_1	Display Changer	execute mode	0		int	W
	s_s	LimitSwitchShaking	Limit Switch	switch	false		bool	R
	s_w	LimitSwitchWashing	Limit Switch	switch	false		bool	R
<input checked="" type="checkbox"/>	a_wash_on	DisplayChangerWater_ON	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_wash_off	DisplayChangerWater_OFF	Display Changer	execute mode	0		int	W
	s_estop	LimitSwitchEStop_1	Limit Switch	switch	false		bool	R
<input checked="" type="checkbox"/>	PWM_counterClockwise	SpeedControlMainMotor_min	Speed Control	speed	-916.200012	*/s	double	W
<input checked="" type="checkbox"/>	PWM_Clockwise	SpeedControlMainMotor_plus	Speed Control	speed	916.200012	*/s	double	W
	PWM_max	Parameters	Expression Block	PWM_max	100.000000		double	R
	PWM_min	Parameters	Expression Block	PWM_min	0.000000		double	R
	Velocity_max	Parameters	Expression Block	Velocity_max	916.200000	*/s	double	R
	Velocity_min	Parameters	Expression Block	Velocity_min	0.000000	*/s	double	R

Signals

A	Name	Data Type	Input/Output	Initial Value	Dimensionality	Unit	Comment
<input checked="" type="checkbox"/>	stopSignal	bool	Output	false			
<input type="checkbox"/>	malfunctionSignal	bool	Output	false			
<input checked="" type="checkbox"/>	limitSwitchStand	bool	Output	false			
<input checked="" type="checkbox"/>	limitSwitchShaking	bool	Output	false			
<input checked="" type="checkbox"/>	limitSwitchWashing	bool	Output	false			
<input checked="" type="checkbox"/>	inductiveSwitchTop	bool	Output	false			
<input checked="" type="checkbox"/>	inductiveSwitchMiddle	bool	Output	false			
<input checked="" type="checkbox"/>	inductiveSwitchBottom	bool	Output	false			
	beaconGreen	bool	Input	false			
	beaconOrange	bool	Input	false			
	beaconRed	bool	Input	false			
	mainMotorEnable	bool	Input	false			
	washMotorEnable	bool	Input	false			
	motorDirection	bool	Input	false			
	pumpEnable	bool	Input	false			
	solenoidLocking	bool	Input	false			
	solenoidStand	bool	Input	false			
	valve	bool	Input	false			
	PWMDutyCycle	int	Input	0			

Formulas

Name

PLCTagsShaker

OK Cancel

Příloha 3 - Adaptér signálů – Formule

Signal Adapter

Parameters

* Select Physics Object (0)

Parameter Name

Add Parameter

A	Alias	Object	Object Type	Parameter	Value	Unit	Data Type	Read/Write
	s_middle	SensorMiddle	Collision Sensor	triggered	false		bool	R
	s_top	SensorTop	Collision Sensor	triggered	false		bool	R
	s_bottom	SensorBottom	Collision Sensor	triggered	false		bool	R
	s_stand	LimitSwitchStand	Collision Sensor	triggered	false		bool	R
<input checked="" type="checkbox"/>	a_capt_1	PositionControlCapt_1	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_capt_2	PositionControlCapt_2	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_lock_1	PositionControlLock_1	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_lock_2	PositionControlLock_2	Position Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_p	SpeedControlWashMotor_plus	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_m	SpeedControlWashMotor_min	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_wm_z	SpeedControlWashMotor_zero	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_p	SpeedControlMainMotor_plus	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_m	SpeedControlMainMotor_min	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_mm_z	SpeedControlMainMotor_zero	Speed Control	active	true		bool	W
<input checked="" type="checkbox"/>	a_red_on	DisplayChangerRed_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_red_off	DisplayChangerRed_OFF_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_orange_on	DisplayChangerOrange_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_orange_off	DisplayChangerOrange_OFF_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_green_on	DisplayChangerGreen_ON_1	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_green_off	DisplayChangerGreen_OFF_1	Display Changer	execute mode	0		int	W
	s_s	LimitSwitchShaking	Limit Switch	switch	false		bool	R
	s_w	LimitSwitchWashing	Limit Switch	switch	false		bool	R
<input checked="" type="checkbox"/>	a_wash_on	DisplayChangerWater_ON	Display Changer	execute mode	0		int	W
<input checked="" type="checkbox"/>	a_wash_off	DisplayChangerWater_OFF	Display Changer	execute mode	0		int	W
	s_estop	LimitSwitchEStop_1	Limit Switch	switch	false		bool	R
<input checked="" type="checkbox"/>	PWM_counterClockwise	SpeedControlMainMotor_min	Speed Control	speed	-916.200012	%s	double	W
<input checked="" type="checkbox"/>	PWM_Clockwise	SpeedControlMainMotor_plus	Speed Control	speed	916.200012	%s	double	W
	PWM_max	Parameters	Expression Block	PWM_max	100.000000		double	R
	PWM_min	Parameters	Expression Block	PWM_min	0.000000		double	R
	Velocity_max	Parameters	Expression Block	Velocity_max	916.200000	%s	double	R
	Velocity_min	Parameters	Expression Block	Velocity_min	0.000000	%s	double	R

Signals

Formulas

Assign to	Formula	Comment
a_capt_1	solenoidStand	
a_capt_2	solenoidStand	
a_lock_1	solenoidLocking	
a_lock_2	solenoidLocking	
a_wm_p	If (washMotorEnable & motorDirection) Then (true) Else (false)	
a_wm_m	If (washMotorEnable & -motorDirection) Then (true) Else (false)	
a_wm_z	If (-washMotorEnable) Then (true) Else (false)	
a_mm_p	If (mainMotorEnable & motorDirection) Then (true) Else (false)	
a_mm_m	If (mainMotorEnable & -motorDirection) Then (true) Else (false)	
a_mm_z	If (-mainMotorEnable) Then (true) Else (false)	
a_red_on	If (beaconRed) Then (1) Else (0)	
a_red_off	If (-beaconRed) Then (1) Else (0)	
a_orange_on	If (beaconOrange) Then (1) Else (0)	
a_orange_off	If (-beaconOrange) Then (1) Else (0)	
a_green_on	If (beaconGreen) Then (1) Else (0)	
a_green_off	If (-beaconGreen) Then (1) Else (0)	
a_wash_on	If (valve & pumpEnable) Then (1) Else (0)	
a_wash_off	If (valve & pumpEnable) Then (0) Else (1)	
PWM_counterClockwise	-(PWM DutyCycle*(Velocity_max-Velocity_min)/(PWM_max-PWM_min))+Velocity_min	
PWM_Clockwise	(PWM DutyCycle*(Velocity_max-Velocity_min)/(PWM_max-PWM_min))+Velocity_min	
stopSignal	s_estop	
limitSwitchStand	s_stand	
limitSwitchShaking	s_s	
limitSwitchWashing	s_w	
inductiveSwitchTop	s_top	
inductiveSwitchMiddle	s_middle	
inductiveSwitchBottom	s_bottom	

Formula

Name

PLCTagsShaker

OK Cancel

Příloha 4 - Obsah přiloženého CD

Zdrojové soubory pro NX 1903 jsou uloženy v adresáři projektu Barman. Řídící programy byly vytvořeny v TIA Portal V15.1.

– ObsluhaMCD.pptx	...	Prezentace pro obsluhu MCD
– Barman_virtual.zap15_1	...	Archivovaný projekt s řídicími programy
– Export	...	Soubor exportovaných signálů
/PLCTagsShaker.xlsx	...	Signály pro MCD
/ShakerModbus.csv	...	Signály pro OPC UA server
/TagsExport.csv	...	Názvy
– Hmotnosti.xlsx	...	Tabulka hmotností
– ShakerRun.mp4	...	Video funkce
– DP_Husak.pdf	...	Elektronická verze diplomové práce