

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA STROJNÍ

Studijní program: N0715A270012 Průmyslové inženýrství a management

**DIPLOMOVÁ PRÁCE**

Návrh a realizace digitálního stínu v Plant Simulation pro existující HW model

Autor: **Bc. Lukáš Veszprémi**

Vedoucí práce: **doc. Ing. Zdeněk Ulrych, Ph.D.**

Akademický rok 2022/2023

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA STROJNÍ

Akademický rok: 2022/2023

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Lukáš Veszprémi**  
Osobní číslo: **S22N0018P**  
Studijní program: **N0715A270012S00 Průmyslové inženýrství a management**  
Studijní obor: **Průmyslové inženýrství a management**  
Téma práce: **Návrh a realizace digitálního stínu v Plant Simulation pro existující HWmodel**  
Zadávající katedra: **Katedra průmyslového inženýrství a managementu**

### Zásady pro vypracování

1. Analýza problému
2. Návrh a vytvoření modelu v Plant Simulation propojený s hardwarovým modelem Fishertechnik včetně 3D animace
3. Návrh a realizace postupu pro přepnutí simulačního modelu na simulaci dlouhodobého chování
4. Ověření navrženého řešení
5. Závěr a zhodnocení

Rozsah diplomové práce: **50–70 stran**

Rozsah grafických prací: **0 výkresů**

Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

1. Bangsow, Steffen. Tecnomatix Plant Simulation: modeling and programming by means of examples, Springer, 2020. ISBN 978-3-030-41544-0.
2. MALAGA, Miroslav a ULRYCH, Zdeněk. Základy řízení robotů pro strojní inženýrství / autoři: Miroslav Malaga, Zdeněk Ulrych [CD-ROM]. První vydání. Plzeň: Západočeská univerzita v Plzni, 2020. ISBN 978-80261-0486-5
3. Mařík, V. Průmysl 4.0: Výzva pro Českou republiku. Management Press, 2016. ISBN 978-807-2614-400

Vedoucí diplomové práce: **Doc. Ing. Zdeněk Ulrych, Ph.D.**  
Katedra průmyslového inženýrství a managementu

Konzultant diplomové práce: **Ing. Bc. Miroslav Malaga**  
Katedra průmyslového inženýrství a managementu

Datum zadání diplomové práce: 4.října 2022

Termín odevzdání diplomové práce: 9.srpna 2023

L.S.

---

**Doc. Ing. Vladimír Duchek, Ph.D.**

děkan

---

**Doc. Ing. Michal Šimon, Ph.D.**

vedoucí katedry

V Plzni dne 4. října 2022

## Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě diplomové práce, zpracovanou v průběhu studia na Fakultě strojní Západočeské univerzity v Plzni. Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této práce.

V Plzni dne: .....

.....

podpis autora

## Poděkování

Rád bych tímto poděkoval svému vedoucímu Doc. Ing. Zdeňkovi Ulrychovi, Ph.D. za odbornou pomoc, cenné rady a vedení při vypracování této práce. Dále bych také velmi rád poděkoval svému konzultantovi Ing. Bc. Míroslavovi Malagovi za ochotu a poskytnutí veškerých potřebných informací.

## ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

<b>AUTOR</b>	Příjmení <b>Veszprémi</b>	Jméno <b>Lukáš</b>	
<b>STUDIJNÍ OBOR</b>	Průmyslové inženýrství a management		
<b>VEDOUcí PRÁCE</b>	Příjmení (včetně titulů) <b>Ing. Ulrych, Ph.D.</b>	Jméno <b>Zdeněk</b>	
<b>PRACOVISŤE</b>	ZČU – FST – KPV		
<b>DRUH PRÁCE</b>	<b>BAKALÁŘSKÁ</b>	<b>DIPLOMOVÁ</b>	Nehodící se škrtněte
<b>NÁZEV PRÁCE</b>	Návrh a realizace digitálního stínu v Plant Simulation pro existující HW model		

<b>FAKULTA</b>	strojní	<b>KATEDRA</b>	KPV	<b>ROK ODEVZD.</b>	2023
----------------	---------	----------------	-----	--------------------	------

### POČET STRAN (A4 a ekvivalentů A4)

<b>CELKEM</b>	66	<b>TEXTOVÁ ČÁST</b>	40	<b>GRAFICKÁ ČÁST</b>	26
---------------	----	---------------------	----	----------------------	----

<b>STRUČNÝ POPIS ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY</b>	Práce je zaměřena na vytvoření modelu, který se propojí s reálným modelem a bude sloužit jako digitální stín. Na simulaci se poté budou zkoušet analyzovat data, která se nejdříve seberou pomocí dlouhodobých simulací. Hlavní přínos je ušetření času sběru dat a vytvoření modelu, na který se bude dále navazovat.
<b>KLÍČOVÁ SLOVA</b>	Digitální stín, digitální dvojče, stavebnice, STEM, HW, řídicí aplikace, model, simulace diskrétní simulace

## SUMMARY OF DIPLOMA SHEET

<b>AUTHOR</b>	Surname Veszprémi	Name Lukáš	
<b>FIELD OF STUDY</b>	Industrial engineering and management		
<b>SUPERVISOR</b>	Surname (Inclusive of Degrees) Ing. Ulrych, Ph.D.	Name Zdeněk	
<b>INSTITUTION</b>	ZČU – FST – KPV		
<b>TYPE OF WORK</b>	<b>BACHELOR</b>	<b>DIPLOMA</b>	Delete when not applicable
<b>TITLE OF THE WORK</b>	Design and implementation of a digital shadow in Plant Simulation for an existing HW model		

<b>FACULTY</b>	Mechanical Engineering	<b>DEPARTMENT</b>	KPV	<b>SUBMITTED IN</b>	2023
----------------	------------------------	-------------------	-----	---------------------	------

### NUMBER OF PAGES (A4 and eq. A4)

<b>TOTALLY</b>	66	<b>TEXT PART</b>	40	<b>GRAPHICAL PART</b>	26
----------------	----	------------------	----	-----------------------	----

<b>BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS</b>	The work is aimed at creating a model that will be linked to the real model and serve as a digital shadow. The simulation will then be used to try to analyse the data, which will first be collected through long-term simulations. The main benefit is to save data collection time and create a model to build on.
<b>KEY WORDS</b>	Digital shadow, digital twin, building block, STEM, HW, control application, model, discrete event simulation

## Obsah

Seznam obrázků .....	10
Seznam tabulek .....	13
Úvod .....	14
1. Digitální dvojče .....	15
1.1 Digitální model .....	15
1.2 Digitální stín .....	16
1.3 Digitální dvojče .....	16
1.3.1 Hodnota propojení a standardizace .....	19
1.3.2 Aplikace ve výrobě .....	19
1.3.3 Další obory využití .....	21
2. STEM stavebnice .....	23
2.1 Fishertechnik .....	23
2.2 VEX .....	24
2.3 LEGO Education .....	25
2.4 Festo .....	26
2.5 Komunikace mezi HW a digitálním stínem .....	27
2.5.1 Sběrnice I <sup>2</sup> C .....	27
2.5.2 MQTT .....	27
2.5.3 Arduino .....	28
2.5.4 RaspberryPI .....	28
2.5.5 Mikro:bit .....	29
3. Simulační program .....	30
3.1 Tecnomatix Plant Simulation .....	30
3.2 Diskrétní simulace .....	31
3.3 Distribuční funkce .....	33
3.4 Řídicí aplikace .....	34
4. Představení reálného modelu .....	37
4.1 Sklad .....	37
4.2 Kontrola .....	37
4.3 Manipulační robot .....	38
4.4 Obrábění .....	39



4.5 Svařování .....	39
5. Návrh a vytvoření modelu.....	41
5.1 Sklad .....	43
5.2 Rentgenová kontrola.....	45
5.3 Vyřazení zmetku.....	46
5.4 Pila.....	48
5.5 Manipulační robot.....	51
5.6 Frézování .....	54
5.7 Vrtání .....	55
5.8 Pec .....	57
5.9 Svařování .....	59
5.10 Třídění a transport výrobků .....	62
5.11 Propojení s modelem .....	64
6. Simulace dlouhodobého chování .....	67
7. Analýza výsledků .....	73
8. ZÁVĚR.....	76
Seznam použité literatury .....	77

## Seznam obrázků

Obrázek 1-1: Tok dat v Digitálním modelu [30].....	15
Obrázek 1-2: Tok dat v Digitálním stínu [30].....	16
Obrázek 1-3: Tok dat v Digitálním dvojčeti [30].....	17
Obrázek 1-4: Příklad užití Digitálního dvojčete [19].....	18
Obrázek 2-5: Ukázka Fishertechnik [12] .....	24
Obrázek 2-6: Ukázka VEX stavebnice [20].....	25
Obrázek 2-7: Ukázka Lego Education [21].....	26
Obrázek 2-8: Ukázka Festo [22] .....	27
Obrázek 2-9: Ukázka Arduino [1].....	28
Obrázek 2-10: Ukázka RaspberryPI [3] .....	29
Obrázek 2-11: Ukázka Mikro:bit [4].....	29
Obrázek 3-12: Ukázka programu Tecnomatix Plant Simulation .....	31
Obrázek 3-13: Vývojový diagram řízení modelu s proměnným krokem [24].....	32
Obrázek 3-14: Rozdělení pravděpodobnosti a distribuční funkce .....	34
Obrázek 3-15: Ukázka řídicí aplikace 1 .....	35
Obrázek 3-16: Ukázka řídicí aplikace 2 .....	35
Obrázek 3-17: Ukázka řídicí aplikace 3 .....	36
Obrázek 3-18: Ukázka řídicí aplikace 4 .....	36
Obrázek 4-19: Sklad.....	37
Obrázek 4-20: Stanoviště kontrola .....	38
Obrázek 4-21: Manipulační robot .....	38
Obrázek 4-22: Stanoviště obráběcích strojů.....	39
Obrázek 4-23: Stanoviště svařování.....	40
Obrázek 5-24: Struktura práce .....	41
Obrázek 5-25: Vymodelované puky .....	42
Obrázek 5-26: Celkový pohled na hlavní frame .....	42
Obrázek 5-27: Grafická struktura modelu skladu .....	43
Obrázek 5-28: Vymodelovaný sklad.....	44
Obrázek 5-29: Ukázka z metody pro vytváření výrobku .....	44
Obrázek 5-30: Vymodelovaná rentgenová kontrola .....	45
Obrázek 5-31: Ukázka použitých metod stroje Rentgenová kontrola.....	46
Obrázek 5-32: Vymodelované vyřazení zmetků .....	47

Obrázek 5-33: Poloha vysunutí .....	47
Obrázek 5-34: Ukázka použitých metod stroje Vyhazovač zmetků .....	48
Obrázek 5-35: Vymodelovaná pila .....	49
Obrázek 5-36: Nastavení animace kotouče pily.....	49
Obrázek 5-37: Nastavení animace dráhy kotouče pily.....	50
Obrázek 5-38: Ukázka použitých metod stroje Pila.....	50
Obrázek 5-39: Manipulační robot .....	51
Obrázek 5-40: Ukázka použitých metod stroje Manipulační robot .....	52
Obrázek 5-41: Grafická struktura modelu manipulačního robota.....	52
Obrázek 5-42: Ukázka polohy manipulačního robota.....	53
Obrázek 5-43: Ukázka nastavení polohy.....	53
Obrázek 5-44: Vymodelovaná frézka .....	54
Obrázek 5-45: Ukázka použitých metod stroje Frézka .....	55
Obrázek 5-46: Animace frézy frézky .....	55
Obrázek 5-47: Vymodelovaná vrtačka.....	56
Obrázek 5-48: Ukázka použitých metod stroje Vrtačka .....	56
Obrázek 5-49: Animace vrtáku vrtačky .....	57
Obrázek 5-50: Vymodelovaná pec .....	57
Obrázek 5-51: Ukázka použitých metod stroje Pec .....	58
Obrázek 5-52: Ukázka kódu metody OnEntrance.....	59
Obrázek 5-53: Vymodelovaná svářečka .....	60
Obrázek 5-54: Ukázka použitých metod stroje Svářečka .....	60
Obrázek 5-55: Grafická struktura modelu svářečky .....	61
Obrázek 5-56: Ukázka pracovní polohy svářečky .....	61
Obrázek 5-57: Ukázka parametrů polohy práce.....	62
Obrázek 5-58: Vymodelovaná potencionální třídička.....	63
Obrázek 5-59: Ukázka kódu ovládání vozíku a vozíku .....	63
Obrázek 5-60: Celkový pohled na model simulace .....	64
Obrázek 5-61: Spuštění aplikace.....	65
Obrázek 5-62: Převodní tabulka z Arduino.....	65
Obrázek 5-63: Nastavení řídicí aplikace .....	66
Obrázek 6-64: Ukázka sběru časů pro jednu barvu puku určitého stroje.....	67
Obrázek 6-65: Použití DataFitu – výběr distribučních funkcí .....	68

Obrázek 6-66: Ukázka doporučení testů .....	68
Obrázek 6-67: Ukázka z výsledků DataFit .....	69
Obrázek 6-68: Ukázka z grafických výsledků DataFit 1 .....	69
Obrázek 6-69: Ukázka z grafických výsledků DataFit 2 .....	70
Obrázek 6-70: Ukázka z grafických výsledků DataFit 3 .....	70
Obrázek 6-71: Ukázka nastavení vstupů pro dlouhodobou simulaci .....	71
Obrázek 6-72: Ukázka nastavení pily pro dlouhodobou simulaci .....	72
Obrázek 7-73: Porovnání času obrábění stanoviště Pila .....	73
Obrázek 7-74: Porovnání využití strojů .....	74

## Seznam tabulek

Tabulka 6-1: Tabulka záznamů časů $t_{Casy}$ .....	67
Tabulka 7-2: Výsledek simulace pro vstupy v kusech .....	74
Tabulka 7-3: Porovnání výsledků experimentů .....	75

## Úvod

V dnešní době, pokud někdo odmítne technologii, s největší pravděpodobností na to doplatí za pár let. Důvodem je to, že konkurence tuto možnost neodmítne a posune se dopředu. Nacházíme se v době, kdy být technologicky vyspělí dává nepředstavitelnou výhodu, a to nejen mezi podniky, ale i např. univerzitami. Mít tu možnost cokoliv monitorovat, sbírat data, a i s nimi pracovat je již na vzestup. Ale co když je potřeba více dat za kratší dobu? Tímto směrem bude právě práce zaměřena, a to na získávání dat za delší časový úsek v krátkém období. Celá práce se bude zabývat na propojení reálného modelu s jeho digitálním stínem, na kterém se budou provádět pokusy.

Teoretická část se zaměřuje především na seznámení s digitálním stínem a dvojčetem. Dále bude psáno o distribučních funkcích, které jsou důležitým faktorem simulací. Poté budou představena STEM stavebnice a její zástupci, kteří se ukazují na trhu. Dalším větším tématem je komunikace mezi hardwarem a softwarem, resp. Digitálním stínem. Dvě větší podkapitoly tohoto tématu jsou protokoly a stroje třetích stran, které se dají programovat.

Začátek druhé teoretické části nám představí samotným programem Tecnomatix Plant Simulation, který bude používán v praktické části. Také bude představena diskrétní simulace, která probíhá v práci.

Propojovací článek mezi teoretickou a praktickou částí bude představení samotného modelu, který se nachází v prostorách Univerzity. Jsou zde představeny jednotlivé části modelu a jejich funkce.

V praktické části je popsáno modelování celé výrobní linie, která právě slouží jako model. Při modelování je popsáno i programování a nastavení částí modelu. Těsný konec je věnován dlouhodobé simulaci, kde budou viděny výsledky analýz s distribučními funkcemi.

Poslední část se zaobírá závěrem.

## 1. Digitální dvojče

Futurologové už léta předpovídají věk "technologické singularity". Ačkoli tento termín obvykle označuje okamžik, kdy umělá inteligence překoná inteligenci lidskou, jiné technologie jsou již mezi námi a stávají se součástí našich životů. Jedním z takových trendů jsou digitální dvojčata, digitální modely a digitální stíny. Problémem, který vznikl v důsledku rostoucí popularity digitálního dvojčete, je nedostatek shody v popisu digitálního dvojčete a také několik různých typů digitálních dvojčat, což přispívá ke zmatku. Proto v následující kapitole budou probrány digitální model, digitální stín a digitální dvojče.

V této práci bude uváděn přístup k digitálnímu dvojčeti dle Tao Fei a Zhang Meng. Digitální dvojče označuje virtuální kopii nebo model jakékoli fyzické entity (fyzické dvojče), které jsou vzájemně propojeny prostřednictvím výměny dat v reálném čase. Konceptně digitální dvojče napodobuje stav svého fyzického dvojčete v reálném čase a naopak. Použití digitálního dvojčete zahrnuje monitorování v reálném čase, navrhování/plánování, optimalizaci, údržbu, vzdálený přístup atd. [30]

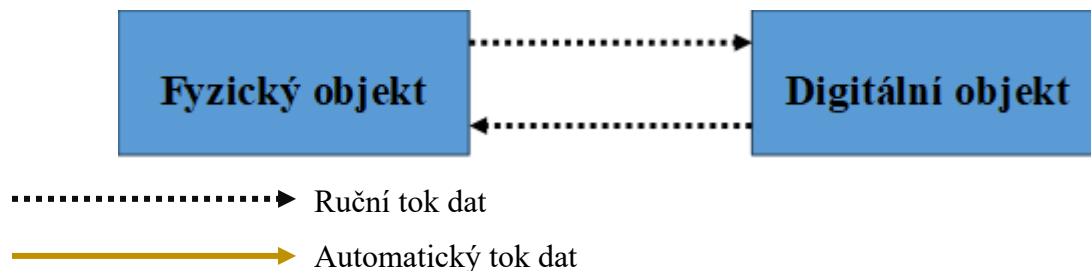
### 1.1 Digitální model

Digitální modely jsou simulací prostředí a lidí v dané oblasti. Digitální model je virtuální reprezentace objektu, kterou lze použít pro simulaci a analýzu.

Koncepční návrh, podrobný návrh objektu, výrobní a konstrukční dokumentace jsou běžná využití CAD. Vyhodnotit alternativy návrhu a analyzovat různé možnosti, aniž by bylo nutné použít skutečné fyzické předměty, je možné pomocí této metody. Před zahájením jakýchkoli prací je třeba modelovat zařízení, materiály a systémy, prověřit 3D plány z hlediska připravenosti

k montáži a stavební fáze a analyzovat ceny. [27]

Tímto postupem se fyzická data přenesou na digitální model. Alternativně může být 3D model jednodušší aproximací nebo podrobným modelem se značným množstvím metadat. Nástroje a množství detailů potřebné k vytvoření takového modelu jsou často dány zamýšleným použitím modelu. Pro návrh schématu nebo výpočty a posouzení stability je vhodná i 2D prezentace. V této fázi projektu nejsou složitosti 3D objektů nutné. Úroveň přesnosti za těchto okolností určuje cíl digitálního modelu. Jednoduchý 3D model pro základní projekt se běžně používá v modelech CAD pro následné využití při důkladnějším projektování a plánování výroby. Jak je vidět na obrázku 1-1, mezi fyzickým a digitálním objektem je tok dat obstarán ručním zadáváním. [30]



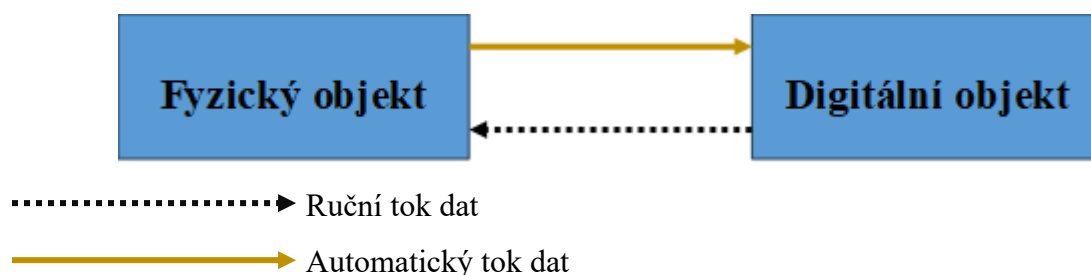
Obrázek 1-1: Tok dat v Digitálním modelu [30]

## 1.2 Digitální stín

Digitální stín je koncept s cílem představit digitální reprezentaci fyzického objektu, systému nebo procesu. Jedná se o virtuální model obsahující kompletní informace o daném objektu, zahrnující jeho vlastnosti, stav, chování a historii. Digitální stín je tedy digitální obraz skutečného objektu. [8]

Hlavním účelem digitálního stínu je umožnit monitorování, analýzu a simulaci daného objektu. Díky tomu je možné provádět různé operace a testování bez nutnosti zasahovat do samotného fyzického objektu. Digitální stín poskytuje informace o stavu objektu v reálném čase, což umožňuje rychlou reakci na události, diagnostiku a predikci chování. [9]

V průmyslových aplikacích se digitální stín využívá pro simulování výrobních procesů, plánování údržby, analýzu provozních dat, simulaci nových scénářů a mnoho dalších úloh. Při implementaci digitálního stínu je nezbytné zajistit správný sběr a aktualizaci dat z fyzického objektu. To často vyžaduje využití senzorů, sběrných systémů a komunikačních protokolů. Důležitou roli hraje také správa dat a jejich integrace s analytickými nástroji a systémy. Jak je vidět na obrázku 1-2, mezi fyzickým a digitálním objektem je sběr dat pro digitální objekt proveden automatizací, ale data zpět do fyzického objektu se musí zadat ručně. [27]



Obrázek 1-2: Tok dat v Digitálním stínu [30]

## 1.3 Digitální dvojče

Digitální dvojče představuje virtuální repliku fyzického objektu, systému nebo zařízení, která je v reálném čase propojena s odpovídajícím fyzickým objektem. Digitální dvojče obsahuje veškeré informace o stavu, chování a vlastnostech fyzického objektu a umožňuje monitorování, simulaci a kontrolu. Digitální dvojče je digitálně vylepšená reprezentace fyzického objektu, například letadla, lodi nebo automobilu. Digitální dvojče bylo vytvořeno tak, aby fungovalo paralelně a komunikovalo s reálným fyzickým výrobním zařízením, čímž se rozšířily možnosti digitalizace počítačem podporovaného navrhování neboli CAD. [30]

Digitální dvojče představuje mocný nástroj, který umožňuje rozšířené možnosti řízení, diagnostiky a analýzy. Jeho hlavním úkolem je poskytovat informace v reálném čase o fyzickém objektu a umožnit efektivní interakci mezi digitálním a fyzickým světem. Díky tomu se dosahuje vyšší přesnosti, rychlosti a efektivity v řízení a provozování zařízení. [9]

V průmyslovém prostředí je digitální dvojče využíváno k monitorování a řízení zařízení a systémů. Pomocí digitálního dvojčete lze simulovat různé scénáře a testovat jejich dopad na fyzický objekt, což umožňuje predikci poruch a optimalizaci provozu. Navíc umožňuje

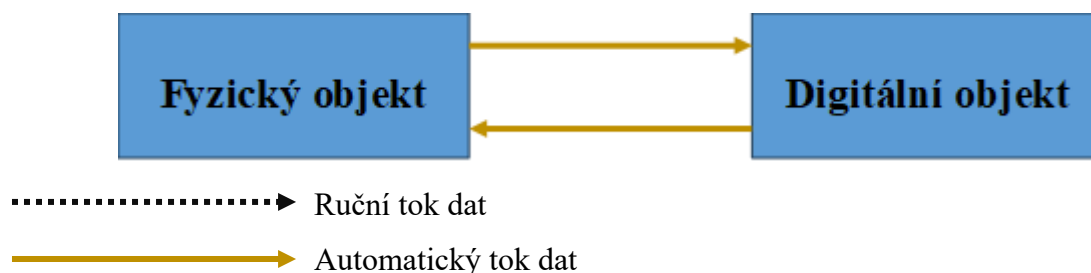


simulovat a analyzovat reakci na změny vstupních podmínek a provádět virtuální experimenty bez nutnosti zasahovat do skutečného objektu. [10]

Implementace digitálního dvojčete vyžaduje integraci dat z fyzického objektu do virtuálního modelu. Sběr dat se provádí pomocí různých technologií, jako jsou senzory, Internet věcí (IoT), sběrnice a komunikační protokoly. Data z fyzického objektu jsou přenášena a zpracovávána v reálném čase, aby byla zajištěna aktuálnost informací v digitálním dvojčeti.

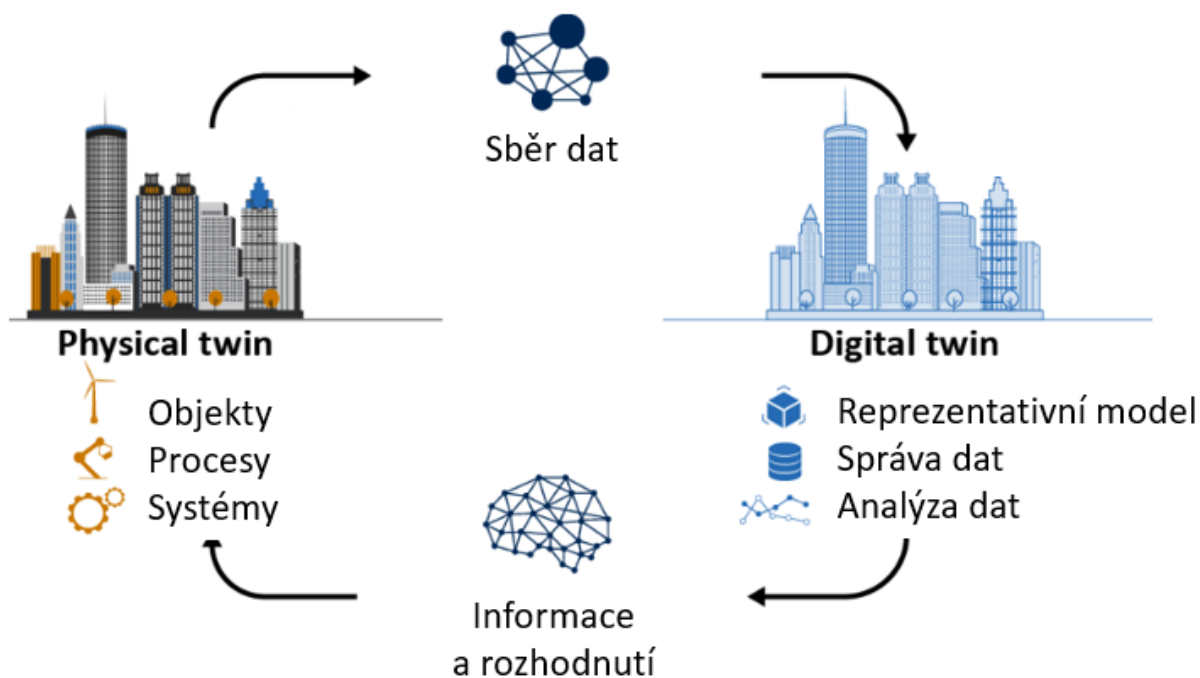
Důležitou součástí digitálního dvojče je jeho schopnost reagovat na události a provádět akce v reálném světě. Toho je dosaženo pomocí propojení mezi digitálním dvojčetem a fyzickým objektem prostřednictvím řídicích systémů a aktivátorů. Tím se umožňuje synchronizace mezi virtuálním a fyzickým světem a realizace akcí založených na informacích z digitálního dvojčete. [11]

Výhody digitálního dvojčete jsou mnohostranné. Poskytuje lepší přehled a pochopení chování fyzického objektu, umožňuje predikci poruch a rychlou diagnostiku. Díky simulacím a testováním lze optimalizovat provozní parametry, plánovat údržbu a zvyšovat energetickou efektivitu. Navíc digitální dvojče umožňuje provádět vzdálenou správu a monitorování objektů, což přináší výhody v oblasti vzdáleného řízení a servisní podpory. Pomocí digitálního dvojčete můžeme zobrazit funkce zařízení, jako by se nacházely ve virtuálním světě. Proto lze ve virtuálním prostředí, které poskytuje, přesně vyhodnocovat složité průmyslové výzvy. Jak je vidět na obrázku 1-3, mezi fyzickým a digitálním objektem je tok dat pro oba směry zajištěn automatizací. [30]



Obrázek 1-3: Tok dat v Digitálním dvojčeti [30]

Jako příklad fungování digitálního dvojčete si můžeme vzít obrázek 1-4. Lze vidět, že jako první se sbírají data z fyzického světa pomocí senzorů na objektech a strojích, které nepřetržitě shromažďují data a posílají je např. pomocí mobilní, drátové nebo bezdrátové sítě. Poté jsou data ukládána např. do cloudových uložišť nebo do databází. Následně digitální dvojče využívá tato data k tomu, aby se dynamicky měnilo spolu s fyzickým objektem v reálném čase. Uživatelé tak mohou interagovat s 3D vizualizací, sledovat změny a předvídat vliv budoucích událostí. Data digitálních dvojčat lze také analyzovat pomocí umělé inteligence a strojového učení k testování různých scénářů nebo k vytváření předpovědí pro fyzická dvojčata. Právě po analýze dat můžeme použít tyto informace pro zlepšení procesů a fungování objektů fyzického dvojčete.



Obrázek 1-4: Příklad užití Digitálního dvojčete [19]

Představme si proto některé analýzy, které můžeme provádět na digitálním dvojčeti:

- Monitorování stavu v reálném čase, které slouží k aktualizaci virtuálního dvojčete v reálném čase.
- Analýza spotřeby energie.
- Analýza a předpověď poruch výrobku a strategie údržby výrobku, které mají společnou analýzu údajů o stavu v reálném čase a historických údajů pro předpověď poruchy a sestavení strategie údržby.
- Inteligentní optimalizace a aktualizace. Tato služba je založena na analýze provozních návyků uživatele a údajů o chování výrobku s cílem zlepšit výrobek a/nebo výrobní proces.
- Analýza chování a průvodce obsluhou uživatele, který slouží k získání informací o operacích prováděných uživateli a/nebo jim poskytuje určité uživatelské pokyny k vizualizaci aktualizací systému pomocí uživatelsky přívětivého rozhraní HMI (Rozhraní člověk-stroj).
- Virtuální údržba výrobku a virtuální operace s výrobkem; s ohledem na 3D prostředí nebo software tyto služby vypracovávají a ukazují uživateli operace nebo strategii údržby. [29]

Představme si některé výhody, které z analýz můžeme získat:

- Optimalizaci výrobních procesů a zároveň snížení rizika nehod a neplánovaných prostojů.
- Zvýšení efektivity s cílem snížit problémy s kvalitou výrobků.
- Využití prediktivní údržby k udržení zařízení a vybavení.
- Testování a ověřování předpokladů.
- Řešení problémů zařízení na dálku.
- Zvyšování produktivity a vytváření efektivnějších dodavatelských řetězců.

- Školení zaměstnanců pro manipulaci se zařízením, které není fyzicky blízko.
- Zlepšování zkušeností zákazníků a pomoc při získávání lepších informací o výkonnosti výrobků.
- Eliminace potřeby více prototypů a zkrácení celkové doby vývoje.
- Aplikace digitálního dvojčete [28] [29]

### 1.3.1 Hodnota propojení a standardizace

Digitální modely, stíny a dvojčata se stále častěji používají ve spojení, aby bylo možné pochopit celý systém a umožnit mu dosáhnout jeho hlavních cílů. Propojení a standardizace digitálních modelů, stínů a dvojčat může v konečném důsledku přinést významné výhody:

- Standardizace  
Standardizace umožňuje bezproblémovou komunikaci různých digitálních reprezentací a integraci napříč oblastmi, systémy a organizacemi. To usnadňuje výměnu dat, spolupráci a interoperabilitu, což vede ke komplexnějším poznatkům a holistickému rozhodování.
- Sdílení znalostí  
Propojení digitálních reprezentací podporuje sdílení znalostí a spolupráci. Umožňuje odborníkům z různých oblastí přispívat svými odbornými znalostmi a poznatky, což umožňuje multidisciplinární řešení problémů. Standardizace rovněž podporuje rozvoj společných rámců, osvědčených postupů a průmyslových standardů, čímž podporuje inovace a efektivitu.
- Škálovatelnost a flexibilita  
Propojení a standardizace digitálních reprezentací vytváří základ pro škálovatelnost a flexibilitu. Organizace mohou snadno replikovat a přizpůsobovat úspěšné modely, stíny nebo dvojčata napříč různými subjekty nebo systémy, což šetří čas a úsilí. Tato škálovatelnost umožňuje použití digitálních reprezentací v široké škále kontextů a podporuje zavádění nových technologií. [28]

### 1.3.2 Aplikace ve výrobě

Výrobní průmysl prochází rychlou transformací. V důsledku toho roste zájem o využití technologií, jako jsou digitální dvojčata, ve výrobním průmyslu. Průmysl 4.0 umožnil technologický pokrok v oblasti snímání, monitorování a rozhodovacích nástrojů. Tyto pokroky napomohly přesné implementaci digitálního dvojčete pro monitorování a optimalizaci procesu v reálném čase. Existuje mnoho potenciálních případů využití digitálního partnerství ve výrobě, včetně monitorování, simulace a vzdáleného ovládní fyzických prostředků pomocí virtuálních objektů. Kromě toho může technologie Digitálního dvojčete pomoci výrobě při zvyšování spokojenosti zákazníků tím, že lépe pochopí jejich potřeby, vyvine vylepšení stávajících produktů, operací a služeb a pomůže při zavádění nových obchodních inovací. Využitím možností Digitálního dvojčete mohou výrobní společnosti přejít od reaktivního přístupu k prediktivnímu. Mohou předvídat, kdy se zařízení opotřebovává nebo potřebuje opravu, zlepšit výkonnost strojů, prodloužit jejich životnost a naučit se, jak je přepracovat, aby dokázaly ještě více. Digitální dvojčata jim navíc umožňují provádět analýzu designu založenou na použití a předprodejní analýzu a přidávat inteligenci do manuálních procesů, aby zlepšily přehled

o potřebách zákazníků atd. Představme si některé oblasti, kde tato technologie může být využita ve výrobním nebo logistickém podniku. [29]

### **Design výrobku**

Zavedení nových produktů nebo služeb může mít dopad na celou organizaci. Kromě toho má design výrobků a služeb strategické důsledky pro úspěch organizace. Rozhodnutí týkající se designu výrobků a služeb proto patří k těm nejzásadnějším, která musí manažeři učinit. Digitální dvojče poskytuje virtuální repliku výrobního prostředku, která shromažďuje data a poskytuje možnost vytvářet, testovat a ověřovat prediktivní analýzy a automatizaci. Inženýři mohou virtuální prototyp vytvořený pomocí digitálních dvojčat využít ve fázi návrhu k testování různých návrhů předtím, než investují do pevného prototypu. Tím se snižuje počet prototypů, šetří se čas a snižují se výrobní náklady. Po rozhodnutí vytvoření prototypu může dojít k dřívějšímu testování a sbírání informací od testerů a případných zákazníků prototypu. [29] Kromě toho mohou inženýři a návrháři využít data shromážděná v průběhu času ke zlepšení očekávání zákazníků, pokud jde o kvalitu výrobku, přizpůsobení a snadné používání.

### **Návrh a optimalizace procesu**

Digitální dvojče pomáhá výrobcům sledovat procesy za různých výkonnostních podmínek a eliminovat problémy dříve, než nastanou. To umožňuje výrobcům přejít od reaktivního k prediktivnímu přístupu. Kromě toho Digitální dvojče pomáhá přeměnit stávající prostředky na nástroje, které optimalizují procesy, šetří peníze a urychlují inovace. [29]

### **Řízení dodavatelského řetězce**

Věčný koloběh rostoucích nákladů dodavatelského řetězce má dopad na hospodářské výsledky všech aktérů. Výrobci, maloobchodníci a distributoři proto označili snižování nákladů na dodavatelský řetězec za klíčové. Vynikající výkonnost dodavatelského řetězce má navíc strategickou hodnotu, která může vést k rychlé finanční návratnosti, často v řádu měsíců, ke zvýšení produktivity a zisku. Technologie digitálního dvojčete může řešit problémy dodavatelského řetězce, včetně výkonnosti balení, správy vozového parku a efektivity tras. Digitální dvojčata navíc mohou pomoci optimalizovat výrobu JIT (just-in-time) nebo JIS (just-in-sequence) a analyzovat distribuční trasy. Tato technologie je užitečná i v dalších důležitých fázích řízení dodavatelského řetězce, včetně vzniku výrobku, vývoje výrobku a distribuce výrobku. [29]

### **Preventivní údržba**

Preventivní údržba se zaměřuje na předvídání, kdy je třeba naplánovat údržbu komponenty nebo systému, aby se snížily náklady a zvýšila provozuschopnost stroje. Digitální dvojčata mohou modelovat jednotlivá zařízení nebo výrobní procesy a identifikovat odchylky, které indikují potřebu preventivních oprav nebo údržby. Cílem je odhadnout, předpovědět, zjistit nebo diagnostikovat stav součásti nebo systému za účelem efektivnější údržby. Tím by se předešlo nákladným poruchám dříve, než dojde k závažnému problému. Mohou také určit, zda lze využít lepší materiály nebo procesy, nebo pomoci optimalizovat dobu cyklu, úroveň zatížení a kalibraci nástrojů. [29]

### **Automatizace řízení projektu**

Digitální dvojčata mohou pomoci se sledováním postupu tým, že poskytují přehled o stavu a výkonnosti projektů v reálném čase. Toho lze dosáhnout integrací dat do virtuálního modelu

z různých zdrojů, jako jsou senzory, drony, laserové skenery a další monitorovací systémy. Řešení Digitálních dvojčat poháněná umělou inteligencí mohou integrovaná data zpracovávat, analyzovat a prezentovat jako průběžný model s denním nebo hodinovým porovnáním s výchozím modelem. Takové sledování by pomohlo řešit běžné problémy s postupem projektu tím, že by včas odhalilo jakékoli odchylky od rozpočtu nebo harmonogramu, což by projektovým týmům zaslalo včas upozornění a umožnilo vypracovat a realizovat potřebné plány obnovy. [29]

### **Plánování zdrojů a logistika**

Významným zdrojem plýtvání může být zbytečný pohyb a manipulace s materiálem, zařízením a pracovní silou. Řešení digitálního dvojčete mohou pomoci toto plýtvání snížit tím, že umožní štihlý přístup k řízení zdrojů. Tato řešení mohou zajistit sledování přidělování zdrojů a sledování plýtvání v reálném čase, a tím zlepšit efektivitu a produktivitu stavebního procesu. Jedním z příkladů využití digitálních dvojčat pro sledování postupu výstavby je integrace dat ze senzorů stavebních zařízení a vozidel. To může v reálném čase poskytovat informace o poloze a využití prostředků a pomoci identifikovat potenciální úzká místa nebo neefektivitu. [29] Dále pokud se bude jednat o autonomní vozidla, např. ve skladu, může digitální dvojče samo upravit trasu vozidla pro větší efektivnost.

### **Kontrola a hodnocení kvality**

Řešení digitálních dvojčat mohou pomoci s kontrolou a hodnocením kvality prací tím, že v reálném čase poskytují přehled o výkonnosti a chování různých součástí a systémů projektu. Technologie digitálních dvojčat může pomocí algoritmů zpracování obrazu analyzovat video nebo fotografické snímky staveniště a posoudit stav různých komponent a materiálů. To může pomoci s odhalením potenciálních problémů nebo závad a s realizací nápravných opatření k zajištění kvality procesu. [29]

### **Spolupráce napříč funkcemi**

Digitální dvojčata se často používají ke sběru provozních dat v průběhu času. Tato data poskytují přehled o výkonnosti, distribuci a zkušenostech koncových uživatelů a mohou je sdílet inženýři, výrobci, prodejci a marketing. Zaměstnanci napříč obory mohou využívat stejná data k přijímání informovanějších rozhodnutí. [29] Naopak digitální dvojče může vyhodnotit, jaké informace potřebuje určitý pracovník a doručit mu je ještě dříve, než si o ně zažádá.

### **1.3.3 Další obory využití**

Zde budou představeny další obory, ve kterých můžeme najít využití digitálního dvojčete.

#### **Chytrá města**

Digitální dvojčata využívají data v reálném čase a poskytují dynamické modely, které ukazují skutečný výkon chytrých měst. To pomáhá zajistit efektivní využívání zdrojů, úsporu nákladů a účinný způsob zavádění udržitelných postupů do stavebnictví. Digitální dvojčata mohou dále nabídnout řešení mnoha složitých problémů, kterým čelí moderní města. Digitální dvojčata měst lze například využít k simulaci plánování katastrof a reakce na mimořádné události během krizí. [29]

### **Maloobchodní prodej**

V maloobchodním sektoru hrají digitální dvojčata klíčovou roli při rozšiřování zákaznických zkušeností tím, že vytvářejí virtuální dvojčata zákazníků. Tato digitální dvojčata pomáhají vytvářet zákazníkům zážitky na míru. [29] Dále mohou pomoci s lepším plánováním v prodejně.

### **Letectví a obrana**

Digitální dvojčata se používají v leteckém a obranném průmyslu ke zvýšení výkonu a zlepšení návrhů výrobků. Národní úřad pro letectví a vesmír (NASA) používá digitální dvojčata k údržbě, provozu a opravám systémů letadel umístěných ve vesmíru. Americká armáda používá digitální dvojčata k potvrzení integrity polovodičů a čipů používaných v munici. [29]

### **Zdravotní péče**

Lékaři mohou bezpečně provádět lékařské zákroky v simulovaném prostředí pomocí digitálních dvojčat orgánu nebo pacienta. Zdravotníci mohou také používat digitální dvojčata k provádění virtuálních klinických testů před uvedením nových vakcín nebo léků na trh. Digitální dvojčata mohou být dále účinným nástrojem pro prediktivní medicínu. Ta může pomoci vyhodnotit riziko pacienta na základě údajů o pacientech s podobným profilem a zaslat preventivně informace doktorovi. [29]

## 2. STEM stavebnice

STEM stavebnice jsou speciálně navržené vzdělávací sady, které slouží k podpoře a rozvoji dovedností ve vědě, technologii, inženýrství a matematice. Zkratka STEM je odvozena z anglických slov Science (věda), Technology (technologie), Engineering (inženýrství) a Mathematics (matematika). Tyto stavebnice mají za cíl zapojit studenty do praktického učení a umožnit jim objevování a aplikaci konceptů a principů ve vědeckých a technických oblastech. [16]

STEM stavebnice poskytují studentům možnost vytvářet, experimentovat a provádět simulace pomocí různých mechanických, elektronických a programovatelných součástek. Nabízejí širokou škálu dílů a modulů, které umožňují konstrukci a tvorbu modelů, jež představují reálné aplikace ve vědě a technologii. [16]

Cílem STEM stavebnic je nejen poskytnout studentům praktický přístup k vědeckým a technickým principům, ale také rozvíjet jejich dovednosti a schopnosti, jako je kreativita, týmová spolupráce, problémové řešení a kritické myšlení. Studenti se aktivně zapojují do procesu učení, kde mají příležitost experimentovat, provádět simulace a aplikovat své znalosti na konkrétní projekty. [17]

STEM stavebnice jsou široce využívány ve vzdělávacích institucích, školách a výzkumných laboratořích, kde představují efektivní nástroje pro praktické a interaktivní vzdělávání. Studenti se prostřednictvím těchto stavebnic učí nejen teoretické koncepty, ale také získávají praktické zkušenosti a dovednosti potřebné pro jejich budoucí kariéru v oblasti STEM. [17]

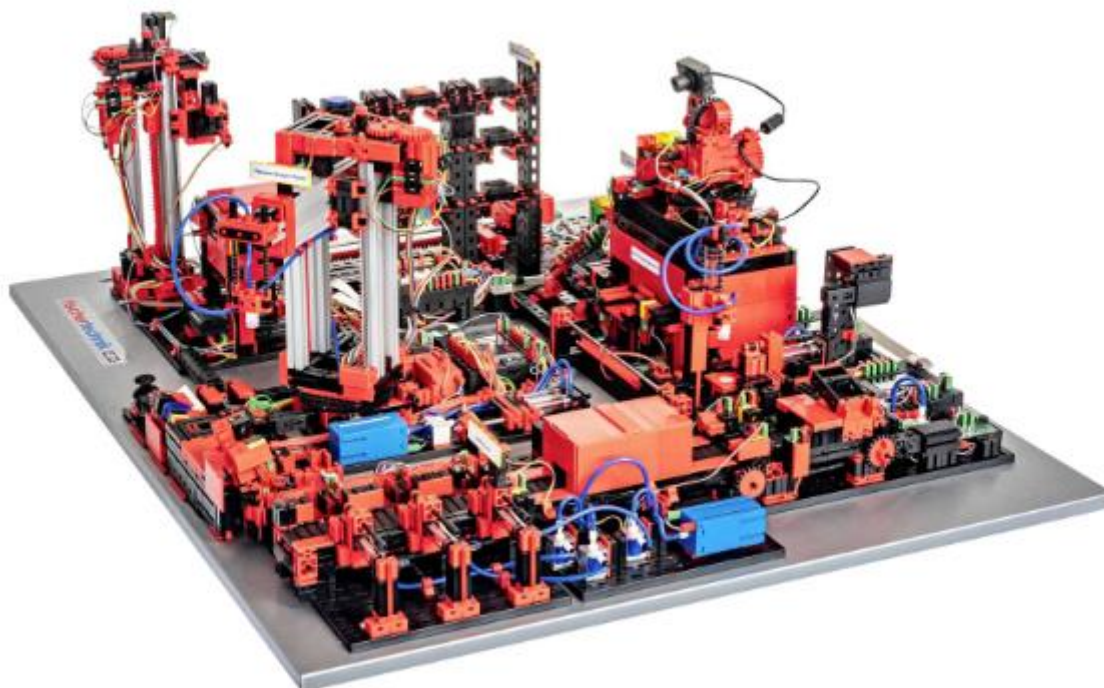
### 2.1 Fishertechnik

Fishertechnik je významnou stavebnicovou sadou, která si získala širokou popularitu jako volba pro výuku a vývoj v oblasti hardwarového modelování na vysokých školách. Jeho výjimečnost spočívá v tom, že umožňuje studentům vytvářet a experimentovat s různými mechanickými a elektronickými systémy. [12]

Fishertechnik je navržen tak, aby poskytoval interaktivní a praktický přístup k učení, což vytváří stimulační prostředí pro objevování a zkoumání technických principů. Jeho modulární design přináší studentům širokou škálu dílů a modulů, které jsou snadno kompatibilní a umožňují rychlou a efektivní konstrukci. Sada obsahuje široké spektrum senzorů, motorů, převodovek a dalších komponentů, které umožňují tvorbu složitějších a funkčních modelů. [12]

Studenti, kteří se zapojí do využití Fishertechnik, mají možnost prozkoumat a simulovat různé technické koncepty, jako je pohyb, síly, elektronika, pneumatika a automatizace. Díky interaktivní povaze stavebnice se studenti aktivně zapojují do procesu učení a rozvíjejí si klíčové dovednosti v oblasti technického myšlení, týmové spolupráce a problémového řešení. [12]

Jeho pověst spočívá v kvalitě, odolnosti a možnostech rozšiřování stavebnice o další komponenty a moduly, které umožňují dále rozvíjet kreativitu a inovace. Fishertechnik tak představuje skvělý nástroj pro praktické a interaktivní vzdělávání v oblasti hardwarového modelování, který podporuje studenty v jejich učebním procesu a připravuje je na technicky náročné výzvy v jejich budoucí profesní dráze. [12]



*Obrázek 2-5: Ukázka Fishertechnik [12]*

## 2.2 VEX

VEX je stavebnicovou sadou, která se zaměřuje na výuku hardwarového modelování, se zaměřením v oblasti robotiky a programování. Tato stavebnice poskytuje studentům skvělou příležitost konstruovat a programovat různé robotické modely a projekty, a tím se prakticky zapojit do procesu učení. [13]

Stavebnice VEX klade důraz na interaktivní a praktické učení. Je vybavena širokou škálou modulů a komponentů, jako jsou motory, senzory, ovládací panely a programovací rozhraní. Tento design umožňuje snadné použití a rychlé sestavování robotických modelů, což studentům umožňuje rychle se dostat k jejich vlastním experimentům a projektem. [13]

S VEXem mohou studenti zkoumat a experimentovat s různými aspekty robotiky, včetně pohybu, sensoriky, programování a komunikace mezi roboty. Stavebnice je také kompatibilní s populárními programovacími jazyky, jako je Scratch, C++ a Python, což umožňuje studentům rozvíjet své dovednosti v programování a algoritmickém myšlení. [13]

VEX je široce využíván v různých vzdělávacích prostředích, včetně škol, technických univerzit a robotických klubů. Jeho modularita a rozšiřitelnost umožňují studentům realizovat své kreativní nápady a rozvíjet své dovednosti a znalosti. Kromě toho, VEX soutěže podporují týmovou spolupráci a soutěživý duch, což přináší studentům inspiraci a motivaci k dosažení lepších výsledků. VEX tak vytváří prostředí, které je pro studenty motivující a nabízí jim zábavný a vzrušující přístup do světa robotiky a technického modelování. [13]





Obrázek 2-6: Ukázka VEX stavebnice [20]

### 2.3 LEGO Education

Lego Education je výjimečnou sadou stavebnic, která přináší inovativní přístup k výuce a podpoře praktického učení. Tato speciální sada je navržena s důrazem na pedagogické účely a umožňuje studentům prozkoumat svět vědy, technologie, inženýrství a matematiky prostřednictvím zábavy a tvůrčích aktivit. [15]

Stavebnice Lego Education je plně vybavena různými moduly, díly a senzory, které umožňují studentům vytvářet a programovat různé modely a projekty. Tímto způsobem se studenti aktivně zapojují do procesu učení a rozvíjejí dovednosti kritického myšlení, týmové spolupráce a řešení problémů. Lego Education podporuje jejich kreativitu a fantazii, a to jak prostřednictvím předem definovaných úkolů, tak prostřednictvím volného tvůrčího projevu. [15]

Kromě samotných stavebních prvků je Lego Education doplněno o speciální software, který umožňuje programování a ovládání vytvořených modelů. Nové verze přechází na programovacího jazyka scratch. Tento software je přístupný a intuitivní pro studenty, ať už jsou na začátku své výuky, nebo mají již pokročilé znalosti programování. Díky tomu se studenti postupně seznámí s principy algoritmizace, sekvenčního a podmíněného programování a mnoha dalších základních konceptů v oblasti automatizace a řízení. [15]

Lego Education je široce využíváno ve školách a vzdělávacích institucích po celém světě. Jeho univerzálnost a rozmanitost umožňuje přizpůsobit výuku různým věkovým skupinám a učebním cílům. Sada podporuje aktivní zapojení studentů, rozvíjení jejich dovedností a posiluje jejich schopnost týmové spolupráce a řešení reálných problémů. Legem Education

se tak stává prostředkem pro inspirativní a interaktivní vzdělávání v oblasti vědeckých a technických disciplín. [15]



Obrázek 2-7: Ukázka Lego Education [21]

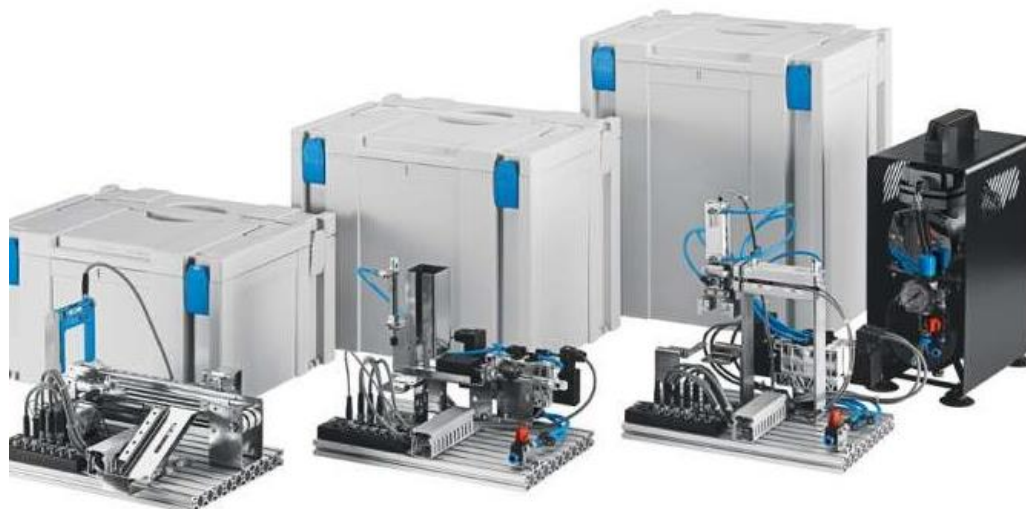
## 2.4 Festo

Festo je uznávaným a renomovaným výrobcem automatizačních technologií a pneumatických systémů. Jeho specializace zahrnuje širokou škálu produktů a služeb, které se používají v průmyslovém prostředí. Společnost Festo se rovněž angažuje v oblasti vzdělávání a nabízí stavebnici nazvanou Festo Didactic, která slouží pro výuku a vývoj v oblasti automatizace a řídicích systémů. [15]

Festo Didactic je navržena tak, aby studentům poskytovala praktický základ a praktické dovednosti v pneumatických, hydraulických a elektrotechnických systémech. Stavebnice obsahuje různé moduly, senzory, ventily a ovládací panely, které umožňují vytváření a simulaci komplexních automatizačních procesů. Cílem je, aby studenti měli možnost seznámit se s reálnými systémy a jejich principy fungování. [15]

Festo Didactic klade důraz na praktické experimentování a simulaci provozních scénářů. Studenti se v rámci výuky seznámí s různými prvky automatizace, jako jsou senzory, řídicí jednotky, PLC (Programmable Logic Controllers) a komunikační protokoly. Stavebnice je doplněna o software, který umožňuje programování a vizualizaci řídicích systémů, čímž umožňuje studentům praktické zkušenosti s programováním a řízením. [15]

Festo Didactic se široce využívá ve vzdělávacích institucích, technických školách a průmyslových společnostech po celém světě. Je to díky jejímu uznání jako kvalitního a spolehlivého vzdělávacího nástroje. Stavebnice Festo Didactic je ceněna pro svoji přesnost, kvalitu a realističnost simulací, které umožňuje studentům získat praktické dovednosti potřebné v oblasti automatizace a řídicích systémů. [15]



Obrázek 2-8: Ukázka Festo [22]

## 2.5 Komunikace mezi HW a digitálním stínem

V této kapitole budou popsány nejdříve dva vybrané komunikační protokoly, a to sběrnice I<sup>2</sup>C a MQTT (Message Queuing Telemetry Transport). Ty mají za cíl výměnu dat mezi objekty. Dále zde budou popsány přídatné zařízení, Arduino, RaspberryPI a Mikro:bit, které mohou např. využívat právě zmíněné protokoly k odesílání dat ze snímačů do digitální fáze nebo k reagování na zprávy z digitální fáze a na základě těchto zpráv iniciovat akce.

### 2.5.1 Sběrnice I<sup>2</sup>C

I<sup>2</sup>C (Inter-Integrated Circuit) je sériový komunikační protokol, který se široce používá pro implementaci hodin reálného času, infračervených senzorů, rozptylovačů a další účely. Sběrnici I<sup>2</sup>C tvoří dvě sériové datové linky a sériová hodinová linka. Arduino může na sběrnici I<sup>2</sup>C pracovat jako master nebo slave. Pro komunikaci s digitálními senzory pomocí sběrnice I<sup>2</sup>C může Arduino fungovat buď jako master, přenášející data získaná ze senzorů připojených k digitálním senzorům, nebo jako slave, přenášející instrukce a provádějící akce na základně zpráv z digitálních senzorů. [2]

### 2.5.2 MQTT

MQTT (Message Queuing Telemetry Transport) je spolehlivý a efektivní komunikační protokol určený pro výměnu dat mezi objekty v reálném světě. MQTT je protokol pro zasílání zpráv založený na standardech nebo souboru pravidel, který se používá pro komunikaci mezi stroji. Chytré senzory, nositelná zařízení a další zařízení internetu věcí (IoT) obvykle musí vysílat a přijímat data prostřednictvím sítě s omezenou šířkou pásma a omezenými zdroji. Tato zařízení internetu věcí používají pro přenos dat protokol MQTT, protože se snadno implementuje a dokáže efektivně komunikovat data internetu věcí. MQTT podporuje přenos zpráv mezi zařízeními do cloudu a z cloudu do zařízení. MQTT využívá architekturu publish-subscribe, kde existuje centrální broker a jsou podporováni klienti pro vydavatele i odběratele. [5]

Z hlediska komunikace mezi hardwarovými a digitálními systémy září MQTT jako odlehčený protokol pro perzistentní data. Hardwarová zařízení, jako je Arduino, Raspberry Pi nebo Micro:Bits, lze nakonfigurovat tak, aby fungovala jako vydavatelé MQTT, kteří odesílají zprávy zprostředkovateli MQTT. Digital Station může fungovat jako odběratel MQTT, který přijímá zprávy, zpracovává je a následně analyzuje. [5]

### 2.5.3 Arduino

Platforma Arduino poskytuje možnosti pro vývoj interaktivních elektronických projektů. Vývody GPIO na desce Arduino umožňují snadnou integraci senzorů, akčních členů a dalších zařízení. Pro komunikaci s digitálními zařízeními je k dispozici několik nástrojů. Jedním z nich je protokol MQTT (Message Queuing Telemetry Transport), který umožňuje zveřejňovat a odevzdávat data zprostředkovatelům v reálném čase. Arduino může fungovat jako klient, který odesílá data ze snímačů do digitální fáze, nebo jako pozorovatel, který reaguje na zprávy z digitální fáze a na základě těchto zpráv iniciuje akce. V současné době je možné se naučit protokol HTTP (Hypertext Transfer Protocol), který umožňuje komunikaci s rozhraními API a webovými službami. Arduino je schopno odesílat požadavky HTTP na digitální polici, která umožňuje ukládat data i časy. [1]



Obrázek 2-9: Ukázka Arduino [1]

### 2.5.4 RaspberryPI

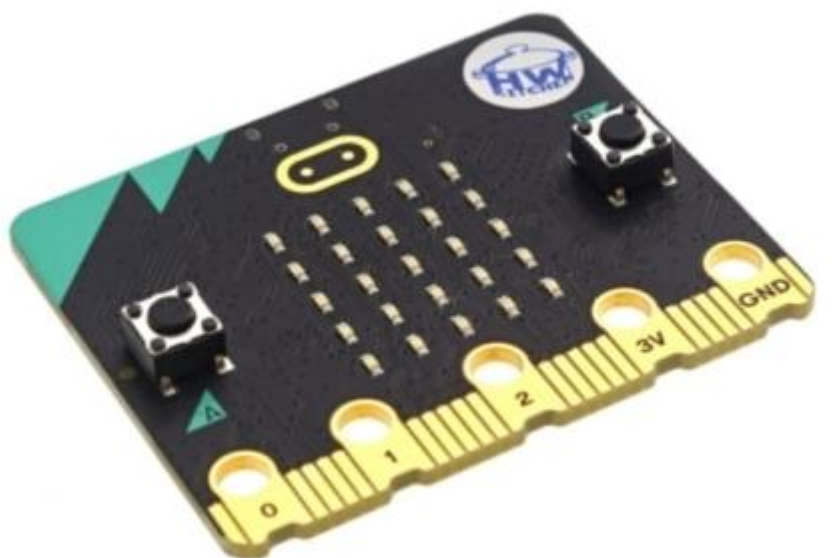
Raspberry Pi je zařízení, které nabízí výjimečné možnosti sestavování hardwaru a komunikace s digitálními přepínači. Raspberry Pi může sloužit jako centrální centrum pro výpočty založené na senzorech, síťové propojení a komunikaci s hodinami reálného času a dalšími procesory. Doporučuje se používat protokoly jako MQTT, který umožňuje publikování a archivaci dat na zprostředkovateli, nebo HTTP, který umožňuje komunikaci s webovými službami a rozhraními API. Raspberry Pi může ukládat data získaná ze snímačů na digitálním hardwaru, stejně jako číst data z těchto snímačů a provádět s nimi operace. [3]



Obrázek 2-10: Ukázka RaspberryPI [3]

### 2.5.5 Mikro:bit

Micro:bit je malé programovací zařízení. Má vestavěné senzory, včetně akcelerometru, teploměru, přepínatelného senzoru a LED displeje. Micro:bit nabízí výkonné možnosti komunikace s digitální pečetí. Má značné použití neomezených technologií, například Bluetooth, které usnadňují komunikaci mezi micro:bitem a dalšími bezdrátovými zařízeními, například digitálními hodinkami. Micro:bit je schopen přenášet data z různých senzorů do digitálního úložiště a ukládat instrukce pro provádění různých operací. K tomu potřebuje micro:bit komunikovat s jinými platformami, jako je Raspberry Pi nebo Arduino, nebo s pomocným hardwarem, jako jsou piny GPIO. [4]



Obrázek 2-11: Ukázka Mikro:bit [4]

### 3. Simulační program

V této části kapitoly bude představen a popsán simulační program Tecnomatix Plant Simulation a jeho využití. Dále, co to je diskrétní simulace, které jsou základem programu Tecnomatix Plant Simulation. Poté bude probrána distribuční funkce, která bude figurovat v praktické části pro vstupy a vyhodnocení. Nakonec této kapitoly bude představena řídicí aplikace, podle které se bude řídit část praktické práce.

#### 3.1 Tecnomatix Plant Simulation

Simulační program Tecnomatix Plant Simulation společnosti Siemens Digital Industries Software se zaměřuje na modelování a analýzu reálných systémů. Jeho konečným cílem je umožnit uživatelům vytvářet virtuální modely robotických linek, upínačů a dalších robotických zařízení a provádět jejich simulace za účelem optimalizace robotického procesu.

Základem Tecnomatix Plant Simulation je koncept diskrétní simulace, který umožňuje přesné modelování a analýzu diskontinuálních operací a procesů v reálném prostředí. Software poskytuje uživatelům bohatou paletu funkcí a vlastností, které jim umožňují zobrazovat virtuální reprezentace reálných systémů a pracovat s nimi.

Pomocí Tecnomatix Plant Simulation mohou uživatelé definovat reálné kroky v daném procesu, včetně strojů, dopravních prostředků, pracovních stanic, skladovacích prostor a dalších. Tyto postupy by měly být navrženy a nakonfigurovány tak, aby poskytovaly skutečné pravdivé výsledky. Je možné provádět simulace, které simulují chování materiálu v kontextu zvoleného modelu.

V simulaci je možné analyzovat přesné parametry daného procesu, včetně takových věcí, jako jsou cykly, stroje, zahřívané materiály a další. Tyto podrobnosti dávají uživateli větší přehled o konotacích a efektivitě jeho rozhodovacího procesu.

Na základě výstupních dat ze simulací mohou uživatelé provádět analýzy a testovat různé scénáře, aby identifikovali slabá místa, úzká hrdla nebo možnosti zlepšení ve svém výrobním procesu. To umožní zavést změny a zlepšení, které zvýší produktivitu a efektivitu.

Tecnomatix Plant Simulation je program, který pomáhá vytvářet digitální modely, které nabízí širokou škálu možností pro modelování, simulaci a optimalizaci reálných systémů. Jeho činnost bude efektivnější a přispěje k lepšímu pochopení výrobních linek a procesů. [6][7]

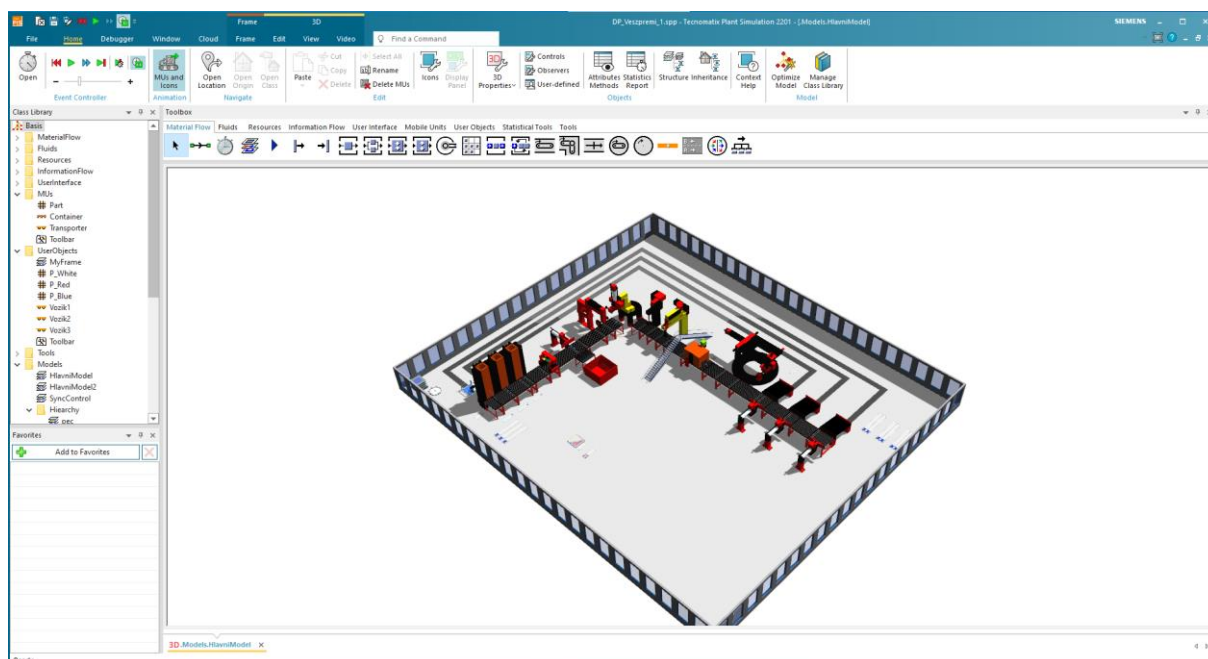
#### **Příklady využití průmyslových odvětví:**

- Automobilový průmysl
- Dodavatelé pro automobilový průmysl
- Letecký a kosmický průmysl
- Průmyslová výroba
- Strojírenství
- Zpracovatelský průmysl
- Elektronický průmysl
- Průmysl spotřebního baleného zboží
- Letiště
- Logistické společnosti (dopravní logistika, skladovací logistika a výrobní logistika)

- Poradenské společnosti a poskytovatelé služeb
- Simulační spolupráce v námořním průmyslu
- Přístavy, zejména kontejnerové terminály

V poslední době nabývá simulace materiálových toků na významu díky stále častějšímu využívání pro zohlednění udržitelnosti průmyslových výrobních procesů. Zde jsou charakteristiky udržitelné výroby simulovány a analyzovány předem a následně začleněny do procesu investičního rozhodování.

Simulace provozu se také používá pro účely výzkumu a vývoje na velkém počtu univerzit a vysokých škol aplikovaných věd. Na obrázku 3-12 lze vidět ukázkou z programu.

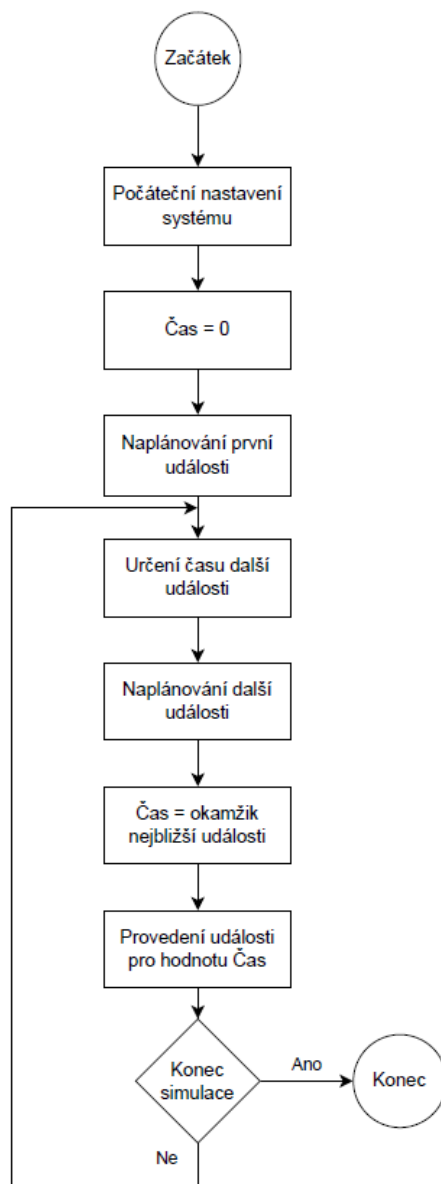


Obrázek 3-12: Ukázkou programu Tecnomatix Plant Simulation

### 3.2 Diskrétní simulace

Diskrétní simulace představuje významný nástroj pro modelování a analýzu systémů, které se vyvíjejí v čase. Tato metoda se zaměřuje na simulaci procesů, které probíhají diskrétně s časovými událostmi odehrávajícími se v diskrétních okamžicích.

Cílem diskrétní simulace je vytvořit model, který přesně popisuje chování systému a umožňuje provádět experimenty a analýzy bez nutnosti fyzického provádění reálných operací. Tímto způsobem lze získat cenné poznatky o systému, optimalizovat procesy, odhalit slabá místa a předpovědět budoucí chování. Na obrázku 3-13 lze vidět vývojový diagram řízení modelu s proměnným krokem



Obrázek 3-13: Vývojový diagram řízení modelu s proměnným krokem [24]

Události jsou klíčovým prvkem v diskrétní simulaci. Představují změny ve stavu systému, které nastávají v konkrétních časových okamžicích. Každá událost je spojena s určitým časem a vyvolává přechod systému z jednoho stavu do druhého. Simulace postupně simuluje jednotlivé události a jejich dopady na systém.

Diskrétní simulace využívá různé techniky a nástroje, mezi které patří například agentní simulace, Petriho sítě, frontové systémy, diskrétní událostní simulace a další. Tyto metody umožňují modelování složitých systémů a analýzu jejich chování v různých scénářích.

V praxi má diskrétní simulace široké uplatnění v různých oblastech, jako je průmysl, doprava, logistika, finanční analýza, výzkum operací, počítačové sítě a další. Poskytuje nástroje pro modelování, optimalizaci a rozhodování v systémech, kde provádění fyzických experimentů je obtížné nebo analytické řešení je příliš složité. [18]



### **Složky diskrétní simulace**

- Čas – Čas plyne skokově.
- Události – Události jsou změny stavu v modelovaném systému (vstup zákazníka do dílny, dodání polotovaru na obráběcí stanici atd.). Lze je popsat změnou stavu entity a okamžikem, kdy ke změně dochází. Ačkoli se události obvykle provádějí v pořadí, některé simulační modely umožňují současné provádění mnoha událostí. Problémem se však stává synchronizace a zaručení správného pořadí událostí.
- Fronta – Čekání na provedení události entitami ve frontě.
- Generátor náhodných čísel – k získávání náhodných čísel se používají generátory pseudonáhodných čísel. Pro napodobení reálných situací, kdy zákazníci vstupují do obchodu v různou dobu a v různém počtu, se právě využívají hodnoty získané z generátorů pseudonáhodných čísel
- Statistika – Výstupem simulace jsou statistické údaje simulovaného jevu, které je třeba dále zpracovat, aby bylo možné získat požadované informace
- Koncové okolnosti – Vzhledem k tomu, že simulace může pokračovat donekonečna, je třeba přidat koncové podmínky, když se simulace ukončí (například po 20 iteracích). [26]

### 3.3 Distribuční funkce

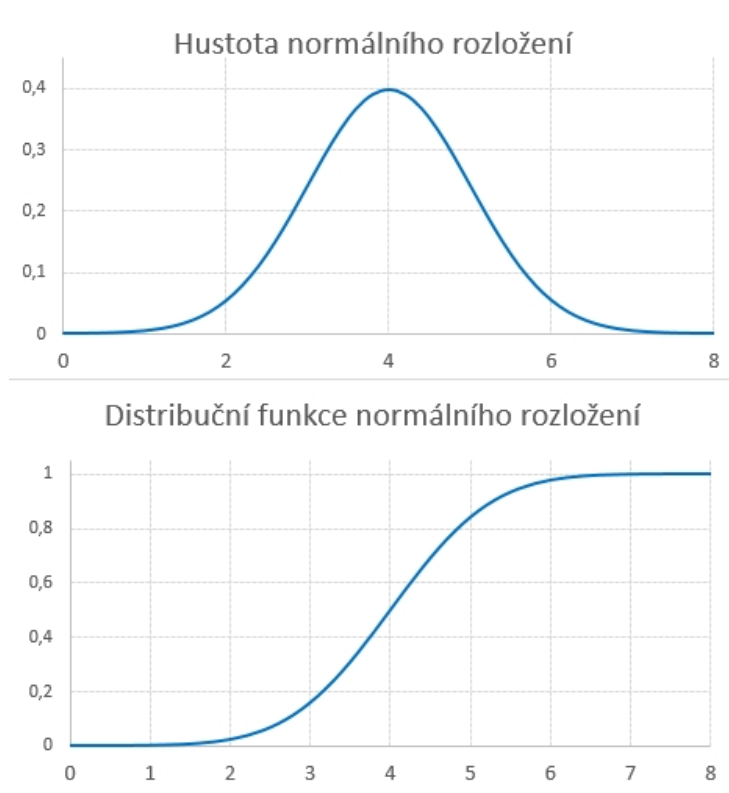
Distribuční funkce se řadí mezi základní koncepty a nástroje v oblasti pravděpodobnosti a statistiky. Jejich úlohou je popsat chování náhodných veličin a poskytnout informace o jejich pravděpodobnostních vlastnostech. Distribuční funkce nám umožňuje analyzovat a modelovat různé jevy v oblasti statistiky, ekonomie, inženýrství a dalších disciplínách.

Úvod do distribučních funkcí nám poskytuje přehled o pravděpodobnostním rozložení náhodných veličin. Distribuční funkce je grafickou reprezentací kumulativní pravděpodobnosti pro různé hodnoty náhodné veličiny. Tato křivka se nazývá kumulativní distribuční funkce (CDF).

Kumulativní distribuční funkce vyjadřuje pravděpodobnost, že náhodná veličina nabývá hodnoty menší nebo rovné určitému bodu. Graficky se kumulativní distribuční funkce zobrazuje jako rostoucí křivka.

Distribuční funkce nám poskytuje různé statistiky o náhodné veličině, jako je průměr, rozptyl, medián, kvartily a percentily. Tyto statistiky jsou klíčové pro analýzu dat a pomáhají nám formulovat závěry a rozhodnutí na základě pravděpodobnostních vlastností.

Existuje mnoho různých distribučních funkcí, které se používají pro modelování náhodných veličin. Mezi nejznámější patří normální (Gaussovo) rozdělení, exponenciální rozdělení, rovnoměrné rozdělení, Poissonovo rozdělení a mnoho dalších. Každé z těchto rozdělení má své charakteristiky a nachází uplatnění v různých oblastech.



Obrázek 3-14: Rozdělení pravděpodobnosti a distribuční funkce

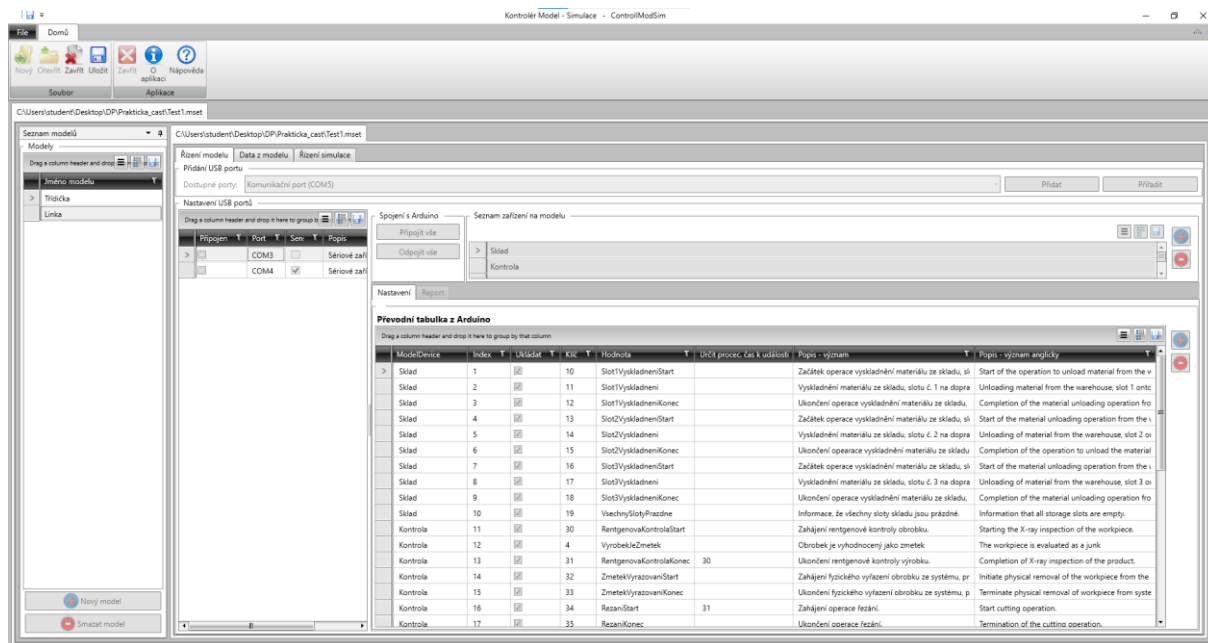
### 3.4 Řídicí aplikace

Řídicí aplikace byla vytvořena aplikací vedoucím diplomové práce, která umožňuje sběr dat z modelu Fishertechnik prostřednictvím řídicí jednotky Arduino. Tato aplikace je navržena tak, aby se připojovala k SQL serveru. Po spuštění aplikace mají uživatelé možnost vytvořit nový model, který je uložen na jejich lokálním počítači. Lokálně jsou uložena pouze nastavení související s komunikací se simulačním modelem. [24]

Samotná data z modelu Fishertechnik jsou ukládána na SQL server. Pokud je pouze potřeba nastavit řízení digitálního stínu pomocí simulačního modelu v Plant Simulation a ovládat tento simulační model z dat uložených na SQL serveru, stačí se do aplikace přihlásit jako host. Tím je umožněn přístup k relevantním informacím a ovládání simulačního modelu bez potřeby ukládání dat lokálně na počítači. Tímto přístupem je zajištěna efektivní správa dat a umožňuje se synchronizace mezi simulačním modelem, řídicí jednotkou Arduino a SQL serverem. [24]

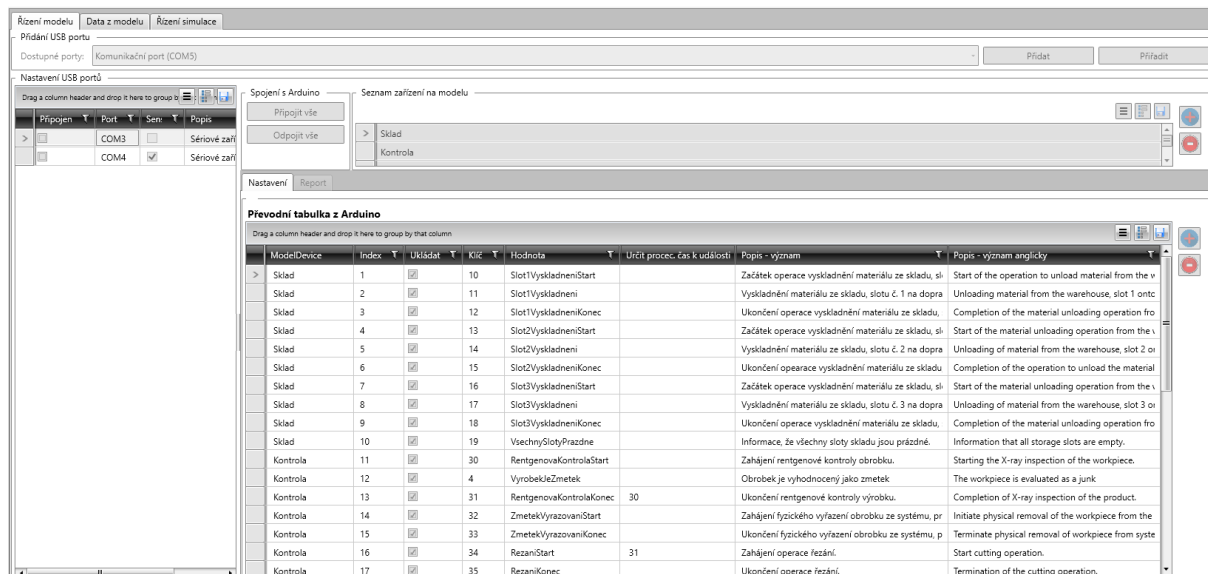
Pro sběr dat z modelu Fishertechnik byla využita originální deska Arduino Uno. Tato deska slouží ke komunikaci s řídicí jednotkou RoboPro TXT a získává informace o prováděných operacích. Kromě toho je možné přímo připojit čidla k desce Arduino Uno, což umožňuje získávat informace o stavu systému přímo z těchto čidel. [24]

Na obrázku 3-15 lze vidět panel po přihlášení se do aplikace. V levé části je sloupec pro výběr z již připravených modelů nebo si zde můžeme vytvořit model nový. V prostřední části jsou tři záložky, a to Řízení modelu, Data z modelu a Řízení simulace.



Obrázek 3-15: Ukázka řídicí aplikace 1

Na obrázku 3-15 je vidět záložka Řízení modelu, kde lze vybrat nastavení USB portů a nachází se zde převodní tabulka z Arduino. Tato tabulka navíc krom hodnot obsahuje i popis významu pro jednotlivé hodnoty jak v českém jazyce, tak anglickém.



Obrázek 3-16: Ukázka řídicí aplikace 2

Další záložkou jsou Data z modelu. Na obrázku 3-17 lze vidět sesbíraná data z fyzického modelu a uložená v databázi.

Simulační čas	Simulační čas [sek]	Zařízení	Hodnota	Čas výroby	Čas výroby [sek]	Data
00:00:44.0431528	44.0431528	Skład	Slot2Vyskladneni	00:00:01.2891875	1.2891875	["dust": 0, "humidity": 21, "light": 95, "noise": 41, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:42.0404258	42.0404258	Skład	Slot2VyskladneniStart	00:00:42.0404258	42.0404258	["dust": 0, "humidity": 21, "light": 95, "noise": 41, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:45.8579140	45.857914	Skład	Slot2VyskladneniKonec	00:00:01.4083951	1.4083951	["dust": 0, "humidity": 21, "light": 95, "noise": 41, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:49.8624449	49.8624449	Kontrola	RentgenovaKontrolaStart	00:00:03.4992640	3.499264	["dust": 0, "humidity": 21, "light": 95, "noise": 41, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:54.4783288	54.4783288	Kontrola	RentgenovaKontrolaKonec	00:00:04.5772155	4.5772155	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:54.8874315	54.8874315	Kontrola	VyrobekNemZmetek	00:00:00.4079336	0.4079336	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:00:57.4832097	57.4832097	Kontrola	RezaniStart	00:00:03.0037042	3.0037042	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:02.8856221	62.8856221	Kontrola	RezaniKonec	00:00:05.4010668	5.4010668	["dust": 0, "humidity": 21, "light": 95, "noise": 39, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:04.4428176	64.4428176	Manipulační robot	ManipulacniRobotStart	00:00:01.5553889	1.5553889	["dust": 0, "humidity": 21, "light": 95, "noise": 39, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:08.7165637	68.7165637	Manipulační robot	UchopenVyrobk	00:00:04.2723305	4.2723305	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:14.9006179	74.9006179	Manipulační robot	UmisteniVyrobk	00:00:06.1826260	6.182626	["dust": 1, "humidity": 21, "light": 95, "noise": 47, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:17.2923622	77.2923622	Manipulační robot	ManipulacniRobotKonec	00:00:02.3894208	2.3894208	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:18.6420054	78.6420054	Obrábění	FrezovaniStart	00:00:01.3477172	1.3477172	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:22.0510483	82.0510483	Obrábění	FrezovaniKonec	00:00:03.4071172	3.4071172	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:23.3496432	83.3496432	Obrábění	VrtaniStart	00:00:01.2970322	1.2970322	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:24.7583684	84.7583684	Obrábění	VrtaniKonec	00:00:01.4073838	1.4073838	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:27.5887867	87.5887867	Svařování	PecStart	00:00:02.8291086	2.8291086	["dust": 0, "humidity": 21, "light": 95, "noise": 38, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:31.9911452	91.9911452	Svařování	PecKonec	00:00:04.3969188	4.3969188	["dust": 0, "humidity": 22, "light": 95, "noise": 39, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:34.3029707	94.3029707	Svařování	SvarovaniStart	00:00:02.3105521	2.3105521	["dust": 0, "humidity": 22, "light": 95, "noise": 39, "pressure": 986, "shakes": 0, "temperature": 26]
00:01:36.638670	96.638670	Svařování	ModyVyrobek	00:00:02.3334189	2.3334189	["dust": 4, "humidity": 22, "light": 95, "noise": 39, "pressure": 986, "shakes": 0, "temperature": 26]

Obrázek 3-17: Ukázka řídicí aplikace 3

Poslední záložkou je Řízení simulace. Na této záložce se vybírá typ licence pro simulační program, nastavení cesty k modelu a nastavení cesty k EventControlleru přímo již v simulačním programu. Dále zde lze vybrat hodnoty, které chceme, aby řídicí aplikace posílala do simulačního programu a zapíná se zde i samotná simulace.

Použití	Zařízení	Hodnota	Cesta k metodě objektu	Parametr metody	Simulační čas další události ze stejného zařízení
<input type="checkbox"/>	Skład	Slot1VyskladneniStart	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot1Vyskladneni	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot1VyskladneniKonec	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot2VyskladneniStart	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot2Vyskladneni	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot2VyskladneniKonec	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot3VyskladneniStart	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot3Vyskladneni	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	Slot3VyskladneniKonec	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Skład	VsechnySlotyPrazdne	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Kontrola	RentgenovaKontrolaStart	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Kontrola	VyrobekjeZmetek	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Kontrola	RentgenovaKontrolaKonec	Station.M_SyncObjectEvent	0	<input type="checkbox"/>
<input type="checkbox"/>	Kontrola	ZmetekVyrzovaniStart	Station.M_SyncObjectEvent	0	<input type="checkbox"/>

Obrázek 3-18: Ukázka řídicí aplikace 4

## 4. Představení reálného modelu

Jako model diplomové práce byl použit model linky, který je vytvořen ze STEM stavebnice Fishertechnik na Fakultě strojů Západočeské univerzity, katedře Průmyslového inženýrství a managementu. Jedná se o model linky, která má představovat výrobní linii v jakékoliv firmě. V modelu je pět hlavních úseků:

### 4.1 Sklad

Sklad je první úsek celého modelu, který simuluje vyskladnění materiálu. Po ručním naplnění alespoň jednoho ze tří zásobníků jedním pukem a zapnutí celého modelu, dojde k vyskladňování jednotlivých barev puků dle algoritmu. Puky najedou na pás a pokračují na další pracoviště. Na obrázku 4-19 lze vidět fotku skladu.

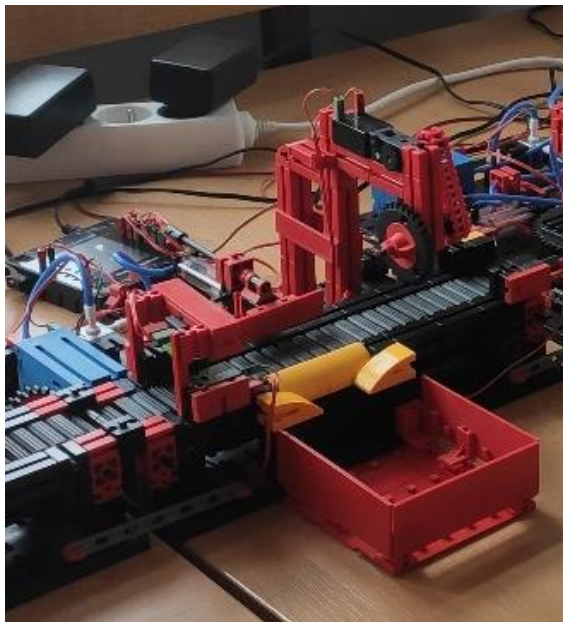


Obrázek 4-19: Sklad

### 4.2 Kontrola

Následující pracoviště po skladu je kontrola. Ta se skládá nejdříve ze stanoviště rentgenové kontroly, které zjišťuje, jestli je výrobek poškozený nebo ne. Tento parametr se již dává do systému při vyskladnění. Poté po pásu výrobek postupuje ke stanovišti vyřazení zmetků. Pokud je puk definován jako zmetek, posuvná ruka ho vyřadí do boční části stanoviště. Pokud puk není vyhodnocen jako zmetek, posouvá se na další část kontroly, kterou je samotná pila. Ta má symbolizovat dělení materiálů. Když puk přijede ke stanovišti, část pily, která symbolizuje

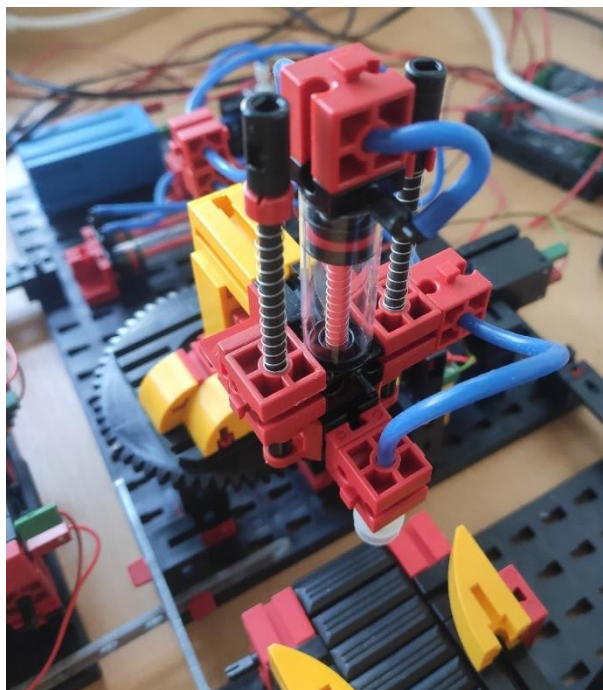
řezací kotouč, začne rotovat a tím se simuluje samotné řezání. Po uplynutí doby řezání dojde puk na konec první části pásu. Na obrázku 4-20 lze vidět fotku skladu.



*Obrázek 4-20: Stanoviště kontrola*

### 4.3 Manipulační robot

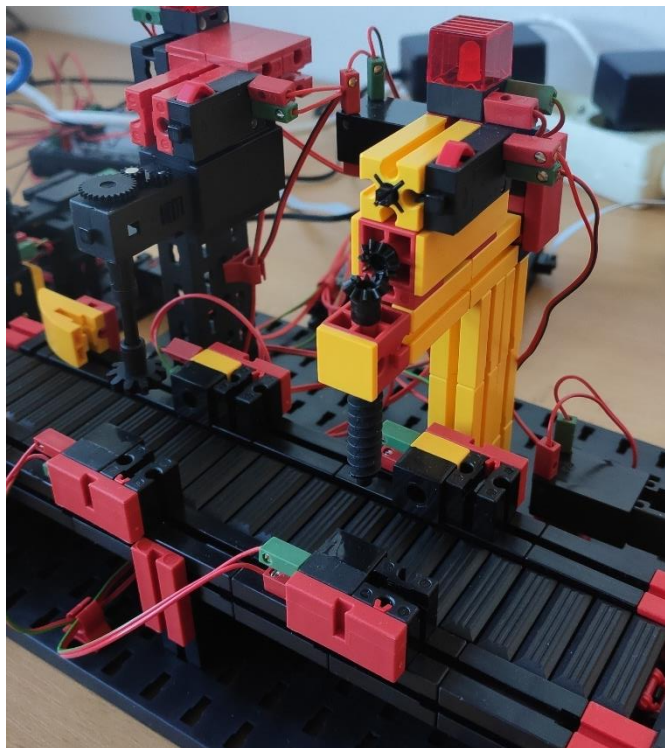
Manipulační robot má za účel přenést puk z prvního pásu na druhý pás. Musí nejdříve rotovat nad puk, který čeká na konci prvního pásu. Poté hlava robota sjede níže nad puk a pomocí přísavky se puk přichytí k robotovi. Následně hlava robota se vyvýší a rotuje nad druhý pás. Posledními kroky jsou snížení hlavy s pukem nad úroveň pásu, jeho položení na pás a zvednutí hlavy do neutrální polohy. Na obrázku 4-21 lze vidět fotku manipulačního robota.



*Obrázek 4-21: Manipulační robot*

#### 4.4 Obrábění

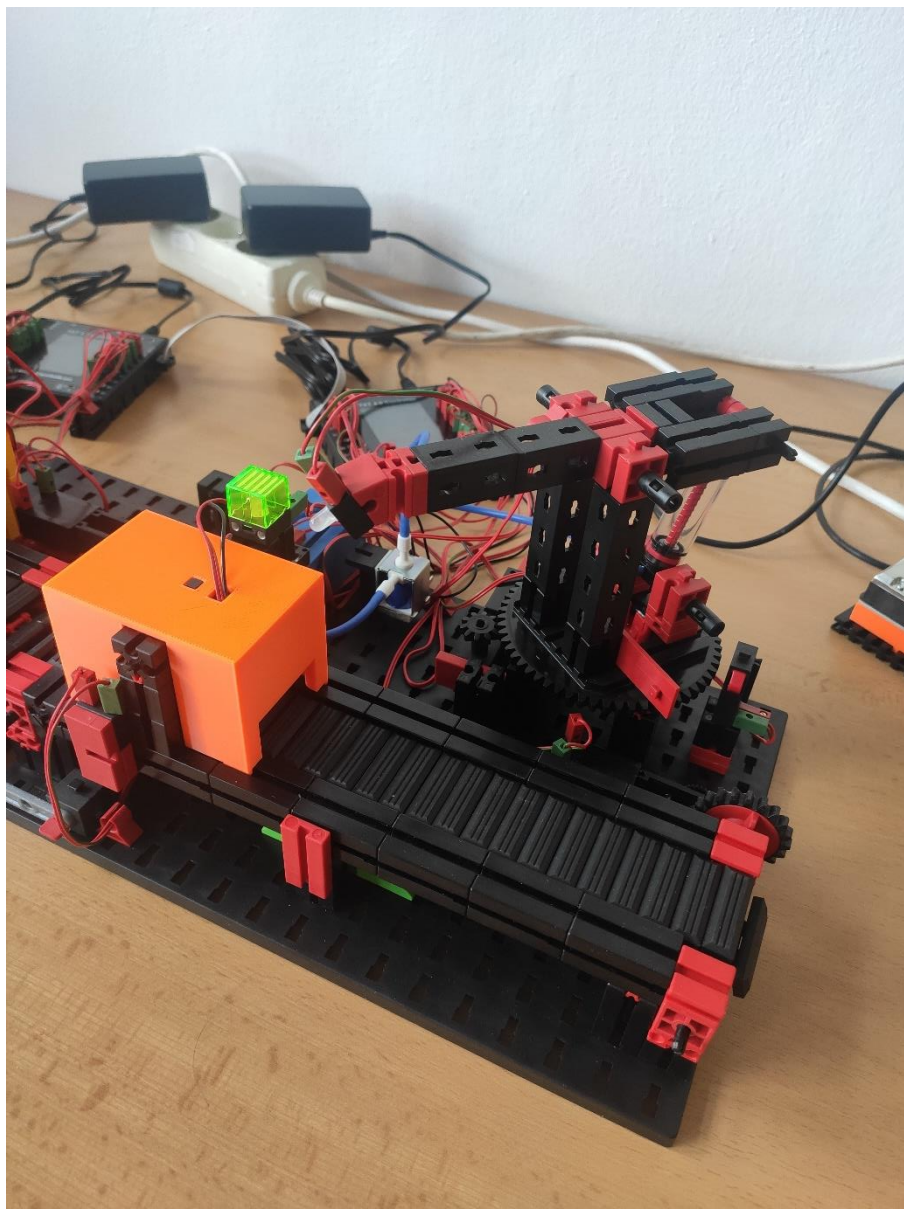
Obrábění se skládá ze dvou obráběcích strojů, frézky a vrtačky. Oba obráběcí stroje mají daný svůj čas obrábění. Navíc každý stroj může simulovat opravu nastavení stroje. To v reálu vypadá tak, že stroj se zastaví a dokud se na stroji nezmáčkne část stavebnice, tak stroj nebude fungovat dále. Pro každý stroj jsou nastaveny obráběcí časy a každý stroj má svoji rotující část. Pro frézku je to samotná fréza, která rotuje kolem své osy. Pro vrtačku je to vrták, který taktéž rotuje kolem své osy. Při obou obráběních je puk pod obráběcím nástrojem. Na obrázku 4-22 lze vidět fotku stanoviště obráběcích strojů.



Obrázek 4-22: Stanoviště obráběcích strojů

#### 4.5 Svařování

Stanoviště svařování se skládá ze dvou částí. První částí je pec a druhou samotné svařování. Do pece vjíždí puk po pásu, kde jsou nastaveny procesní časy, která simuluje materiálové změny. Také se zde zjišťuje barva puku, která je důležitá pro další pracoviště, svařování. Samotná svářečka má nastavené jiné procesní časy pro každou barvu puku. Proto je nutné, aby ze stanoviště pece již byla známa barva. Celá svářečka začíná směrem od pásu na bok. Když přijede puk po pásu, svářečka rotuje tak, aby hlava byla nad pásem. Poté se celá nakloní tak, že právě hlava je nad samotným pukem. Po dokončení samotné simulace svařování se hlava vrátí do původní polohy a poté i celá svářečka, tedy do boční polohy. Na obrázku 4-23 lze vidět fotku stanoviště svařování.

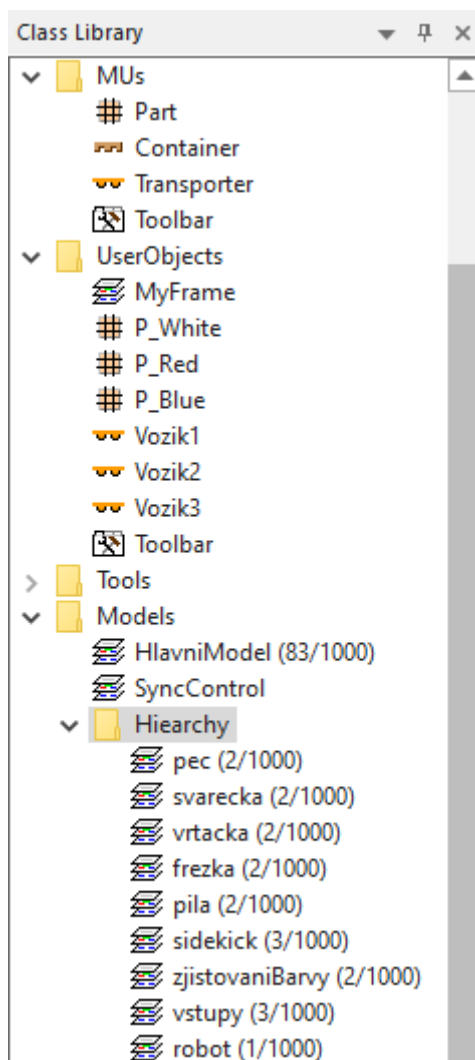


*Obrázek 4-23: Stanoviště svařování*



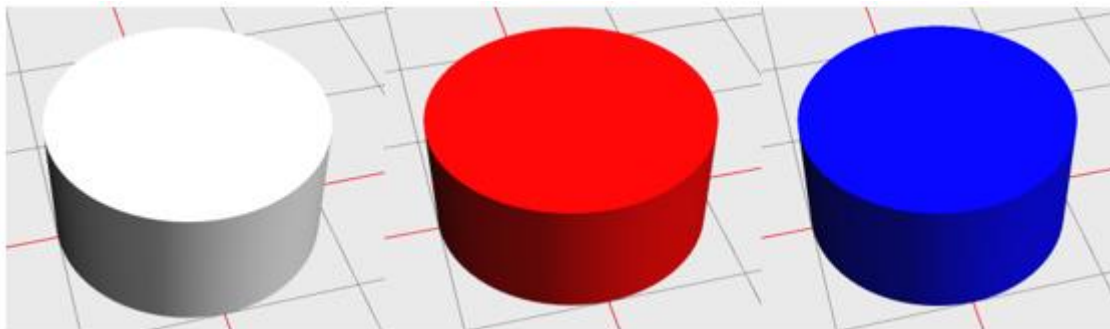
## 5. Návrh a vytvoření modelu

Po založení souboru byla jako první navrhována struktura pro lepší přehlednost celé práce a pro budoucí potencionální použití modelu. Navrhovanou strukturu můžeme vidět na obrázku 2-20. Jako první byl založen frame HlavniModel. V tomto modelu se nachází všechny části projektu, strukturně se jedná o nejvyšší úroveň modelu. Pro jednotlivé stroje byl založený zvlášť frame. Poslední frame je SyncControl. Frame SyncControl byl použit od vedoucího diplomové práce, kterou naprogramoval. Má za účel propojení právě naprogramované aplikace na sběr dat z reálného modelu s programem Tecnomatix Plant Simulation.



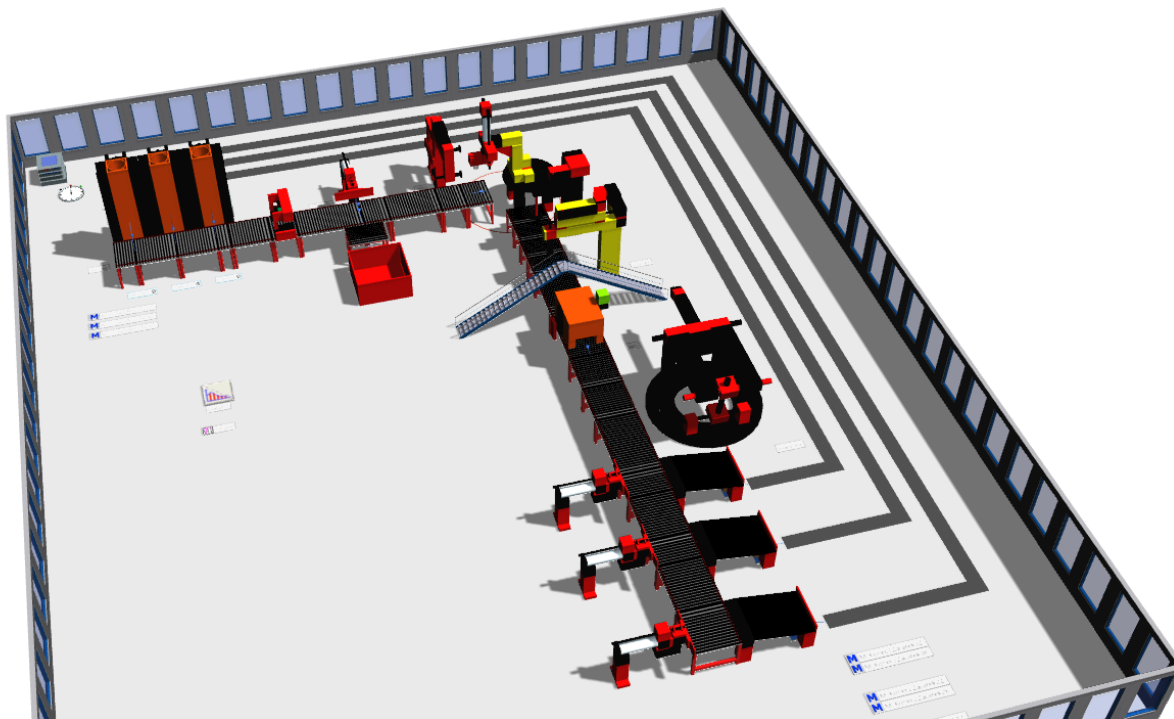
Obrázek 5-24: Struktura práce

Dále ve složce UserObjects byly vytvořeny 3 Parts (výrobky), které jsou pojmenovány jako P\_White, P\_Red a P\_Blue, a 3 Transporters (vozíky), které jsou pojmenovány Vozik1, Vozik2 a Vozik3, které lze vidět na obrázku 5-24. Parts (výrobky) byly upraveny do tvaru puku jako v reálu. Dále byly dány 3 barvy, a to bílá, červená a modrá, které lze vidět na obrázku 5-25. Transporters (vozíky) byly vybrány ze základní knihovny programu pro budoucí návrh, jak by se výrobní linka mohla rozšířit.



*Obrázek 5-25: Vymodelované puky*

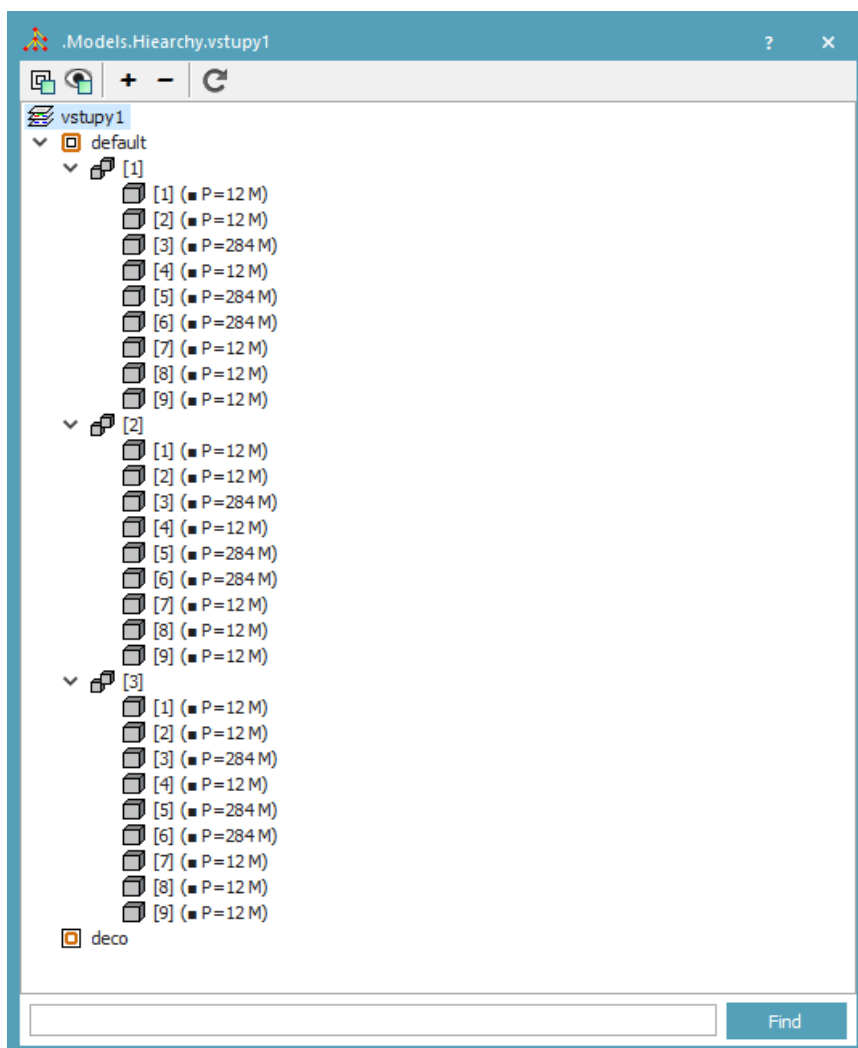
Jako podlaha prostředí byla zvolena bílá deska. Pro okolí modelu byl vytvořen model zdi s okny. Všechny stroje a části, které byly vytvořeny, byly vymodelovány pouze v programu Tecnomatix Plant Simulation. To by nám mělo zaručit bezproblémovou funkčnost modelu. Proto také stroje nejsou v takové grafické kvalitě, jako kdyby byly modelovány v programech k tomu určeným. Zde byly použity pro modelování pouze základní geometrické tvary, jako je kvádr, válec, jehlan a koule. Pro budoucí práci se stroji je to naopak výhodou modelovat stroje v programu, protože se lépe upravují jednotlivé části a poté i lépe animují. Všechny jednotlivá stanoviště jsou propojena pásy, kde byla změřena rychlost posunu a nastavena pro všechny pásy. Celkový pohled na model lze vidět na obrázku 5-26.



*Obrázek 5-26: Celkový pohled na hlavní frame*

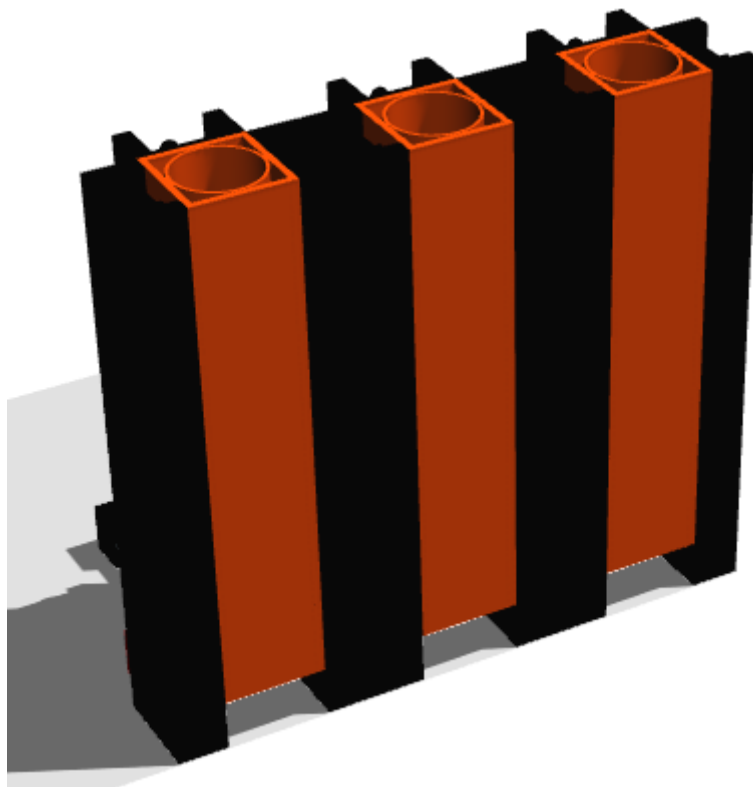
## 5.1 Sklad

Pro vymodelování skladu byly použity buffery, kde se po propojení s aplikací generují výrobky. Samotné buffery byly vymodelovány samostatně, kdy jejich grafická část byla celá předělána.



Obrázek 5-27: Grafická struktura modelu skladu

Na obrázku 5-27 můžeme vidět grafickou strukturu skladu. Jsou zde vymodelovány celkově 3 identické sloupce, kde každý z nich obsahuje devět prvků. Tyto prvky představují základní geometrické útvary a byly vybrány čistě z nabídky programu Plant Simulation.



Obrázek 5-28: Vymodelovaný sklad

Na obrázku 5-28 je již vidět 3D pohled na model skladu. Ten vychází právě z grafické struktury modelu skladu, která je vidět na obrázku 2-23. Jednotlivým prvkům byla vybrána barva a lesk.

Naprogramováno bylo 6 metod, a to Init, M\_OnExit, OnExit, M\_CreateMU\_Slot1, M\_CreateMU\_Slot2 a M\_CreateMU\_Slot3. První zmíněná, Init, pouze přikazuje zamknutí exitu bufferu ihned po zapnutí simulace. Metoda M\_OnExit odemkne buffer po přijetí hodnoty (pro Buffer 1 je hodnota Slot1Vyskladneni) z řídicí aplikace a jakmile se objeví výrobek uvnitř, nastaví se čas další události z parametru metody. Poslední metoda zamyká opět výstup po přijetí signálu z řídicí aplikace (pro Buffer 1 je hodnota Slot1VyskladneniKonec). Zbývající 3 metody vytvářejí výrobek v Bufferech 1, 2 a 3. Jejich funkce je vytvářet výrobek. Tento postup je popsán pod obrázek 5-29 jako ukázka kódu. Dále tato metoda zapisuje čas vytvoření výrobku do tabulky t\_Casy pro budoucí analýzu sebraných časů a jejich vyhodnocení. Tento postup byl aplikován na 3 buffery, kde se generují výrobky dle barvy, a to tedy bílá, červená a modrá. Pro každý buffer byla napsána navíc metoda, která právě generuje dle události uvnitř bufferu výrobek dle barvy. Dále přiřadí výrobku atribut zmetku, který bude sloužit do budoucna na dalším stanovišti. Dalším vytvořeným atributem je položka, kam se ukládá čas další události.

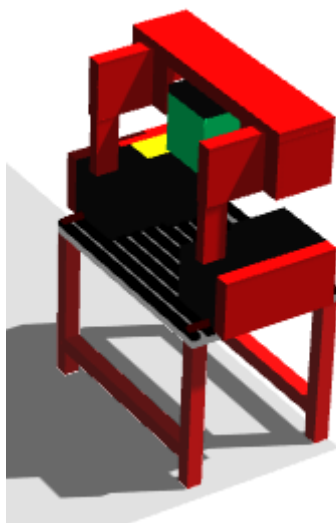
```
param parameter : real, nextEventTime : real
var item: object
item := .UserObjects.P_White.create(B_White)
item.createAttr("NextEventTime", "Time")
item.createAttr("Zmetek", "Boolean")
item.NextEventTime := nextEventTime
```

Obrázek 5-29: Ukázka z metody pro vytváření výrobku

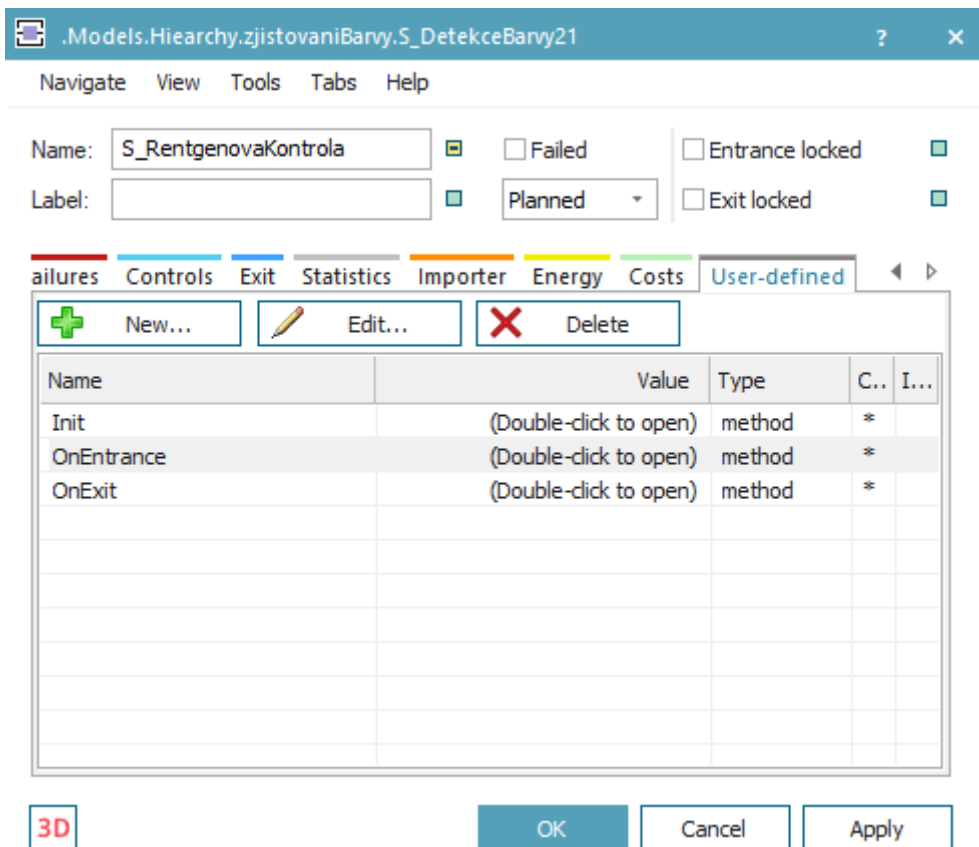
Na obrázku 5-29 můžeme vidět ukázkou z metody pro vytváření výrobku. Metoda popisuje, jak se vytváří bílý výrobek P\_White v bufferu B\_White. Vytvoří se atribut NextEventTime s parametrem Time, který odpovídá času, a atribut Zmetek s parametrem Boolean, který odpovídá hodnotám ano nebo ne.

## 5.2 Rentgenová kontrola

Rentgenová kontrola je stanice, která byla opět namodelována ze základních geometrických útvarů. Na obrázku 5-30 lze vidět vymodelovaný stroj. Zde byly použity 3 metody pro řízení stroje. První metodou je opět Init, který po zapnutí simulace zavírá, jak vstup, tak výstup stroje. Druhou metodou je OnEntrance. Po získání určité hodnoty z řídicí aplikace pro otevření rentgenové kontroly, se otevře vstup a výrobek vstupuje do stroje. Na základě rozdílů kroků pro vstup výrobku do pece a výstup výrobku z pece, které se řídí podle přijatých hodnot z řídicí aplikace, metoda dopočítá procesní čas. Podle barvy, která je určena v parametru každého výrobku, se zapíše tento procesní čas do tabulky t\_Casy, která je rozdělena dle barev výrobků a dle jednotlivých stanovišť strojů. Třetí metodou je metoda onExit, která po přijetí hodnoty z řídicí aplikace uzavře výstup. Na obrázku 5-31 lze vidět použité metody pro pracoviště rentgenové kontroly.



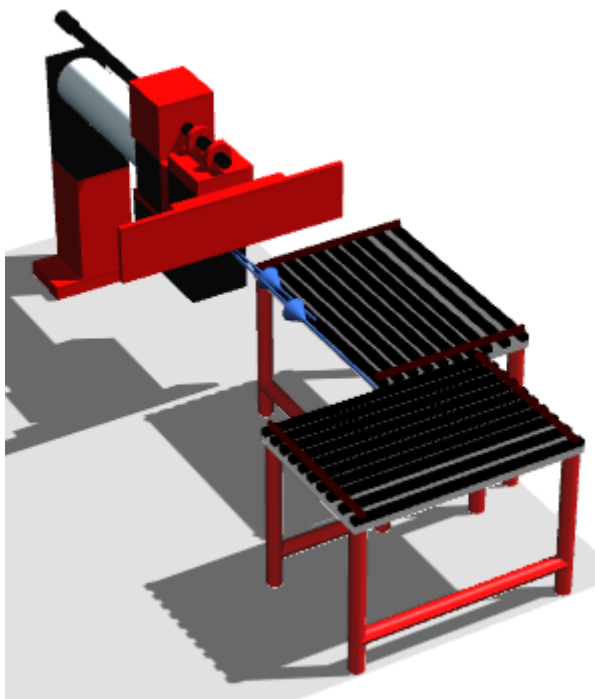
Obrázek 5-30: Vymodelovaná rentgenová kontrola



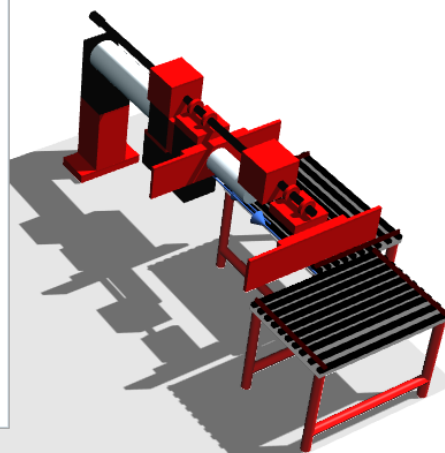
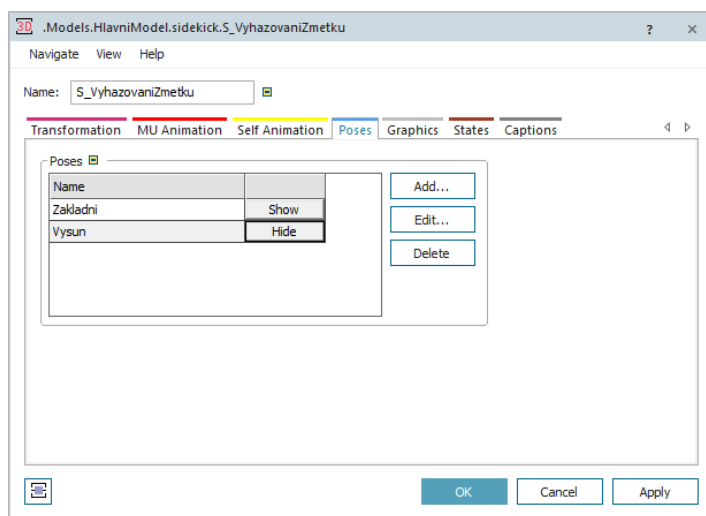
Obrázek 5-31: Ukázka použitých metod stroje Rentgenová kontrola

### 5.3 Vyřazení zmetku

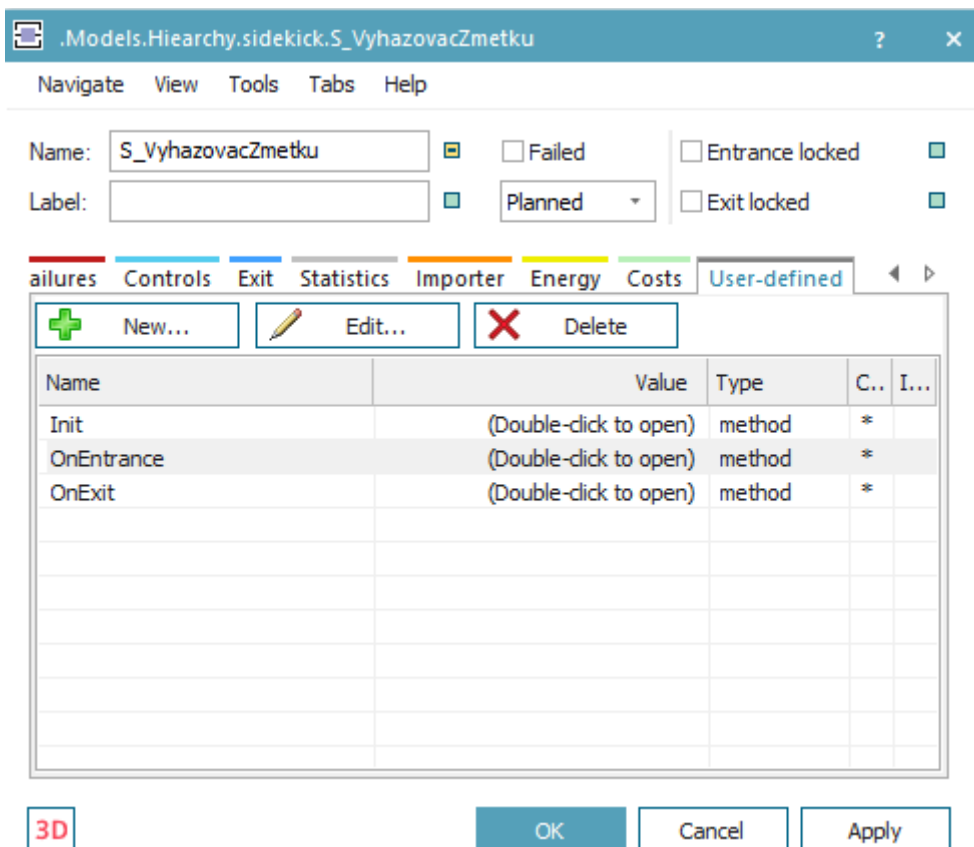
Další stanoviště vyřazuje zmetky pryč z výrobní linie, které lze vidět vymodelované na obrázku 5-32. První metodou je opět Init, který po zapnutí simulace zavírá, jak vstup, tak výstup stroje. Na začátku celé výrobní linie každý výrobek dostane vlastní atribut „Zmetek“, podle kterého se právě na tomto stanovišti určuje, zdali výrobek pojedle dále na další pracoviště nebo bude vyřazen. To nám zaručuje metoda OnEntrance. Při vyřazování navíc proběhne animace vysunutí pístu, který symbolizuje posunutí výrobku na vedlejší pás. Animace je provedena pomocí poloh, které lze vidět na obrázku 5-33, které se musí vymodelovat a nastavit jejich rychlost. Poté byla přiřazena do metody OnEntrance, kdy se vyřazuje výrobek. Třetí metodou je metoda onExit, která po přijetí hodnoty z řídicí aplikace uzavře výstup.



Obrázek 5-32: Vymodelované vyřazení zmetků



Obrázek 5-33: Poloha vysunutí

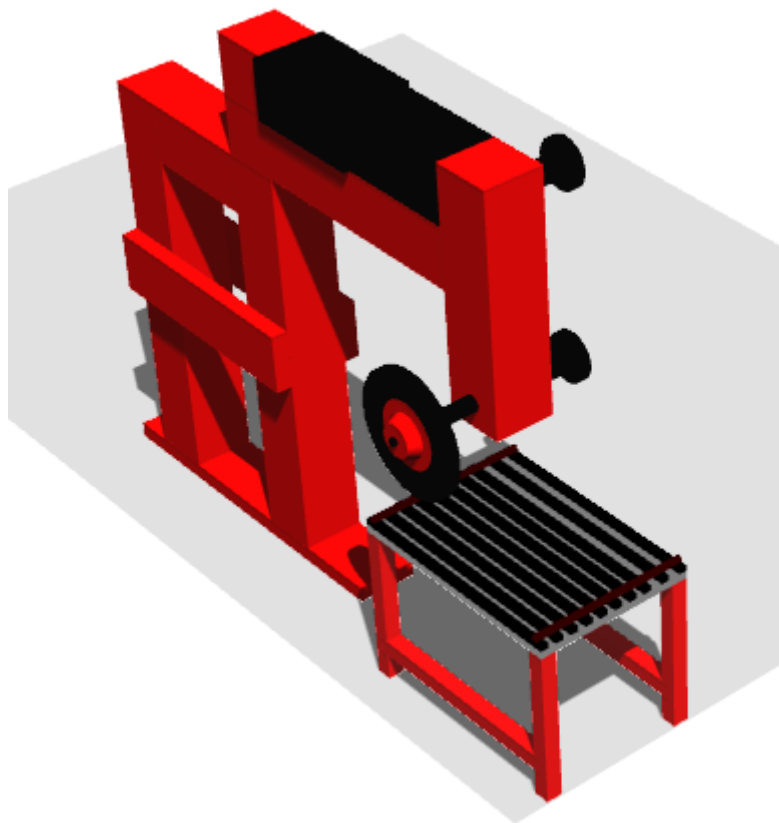


Obrázek 5-34: Ukázka použitých metod stroje Vyhazovač zmetků

## 5.4 Pila

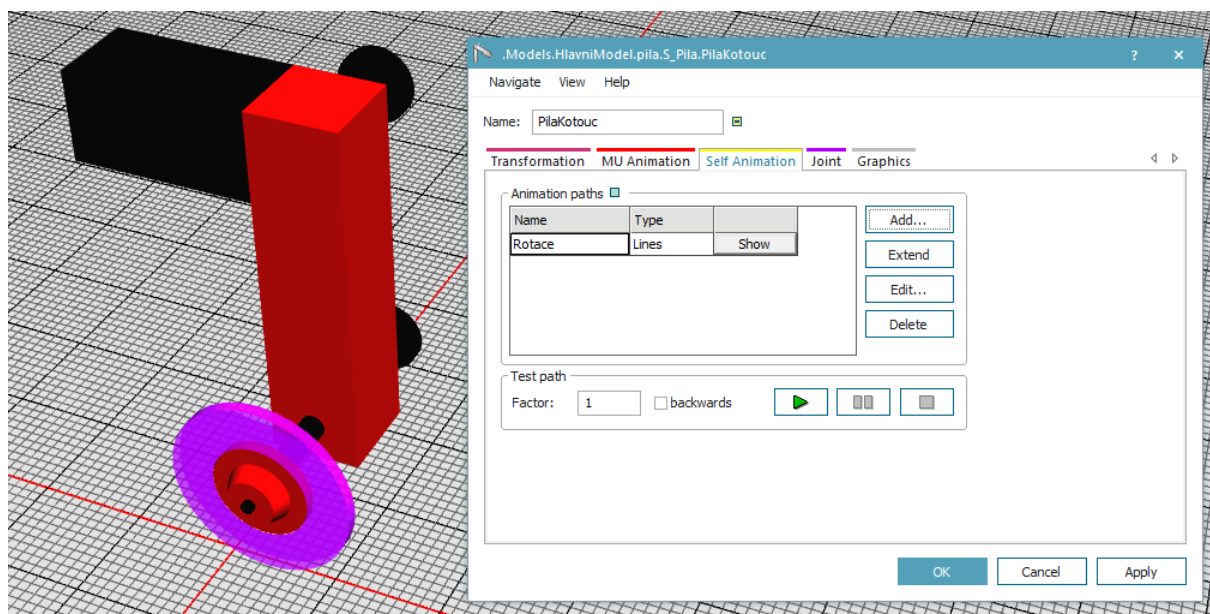
Dobré kusy dojedou po pásu na pracoviště pila, která je vymodelována na obrázku 5-35. Zde opět funguje metoda Init, která po spuštění programu zavírá, jak vstup, tak výstup. Dále pokud přikáže řídicí aplikace pomocí hodnot určených pro pilu, otevře se vstup, dopočítá se procesní čas a provede se samostatný proces, to vše pomocí metody OnEntrance. Při tomto procesu je zde animace rotace kotouče pily. Ta je nastavena na délku procesu v metodě onEntrance. Animace probíhá rotací zmíněného kotouče kolem své osy. Dále probíhá zapisování procesních času do tabulky t\_Casy dle rozdělení jednotlivých barev. Po přijetí určité hodnoty z řídicí aplikace pro posun výrobku se výrobek posouvá dále po pásu a výstup se uzavírá pomocí metody OnExit.



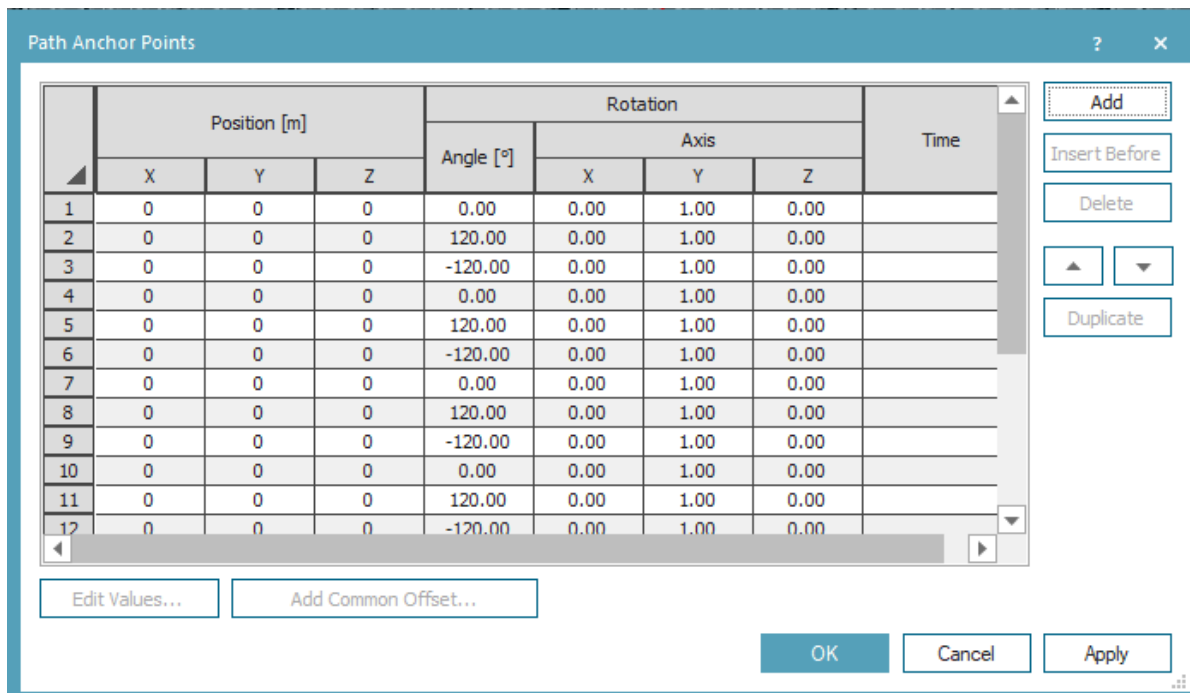


Obrázek 5-35: Vymodelovaná pila

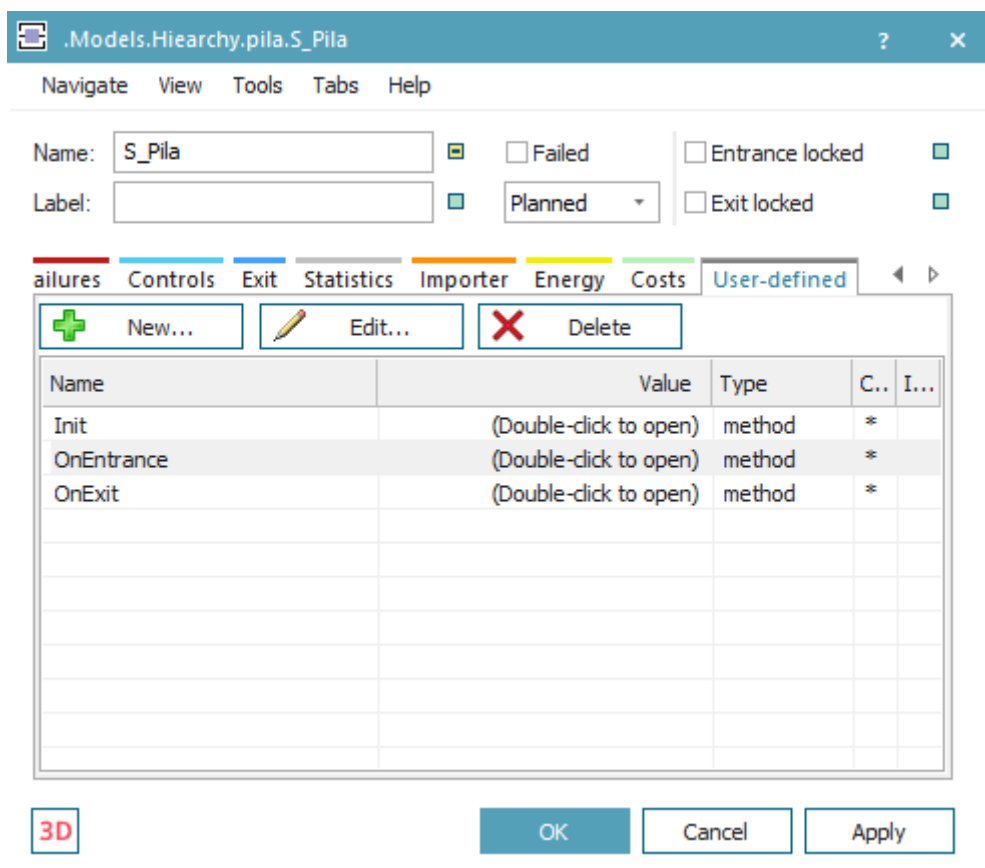
Na obrázku 5-36 můžeme vidět nastavení animace kotouče pily. Jako hlavní animační pohyb zde byla vybrána rotace. Jednotlivé souřadnice bodů dráhy animace můžeme vidět na obrázku 5-37.



Obrázek 5-36: Nastavení animace kotouče pily



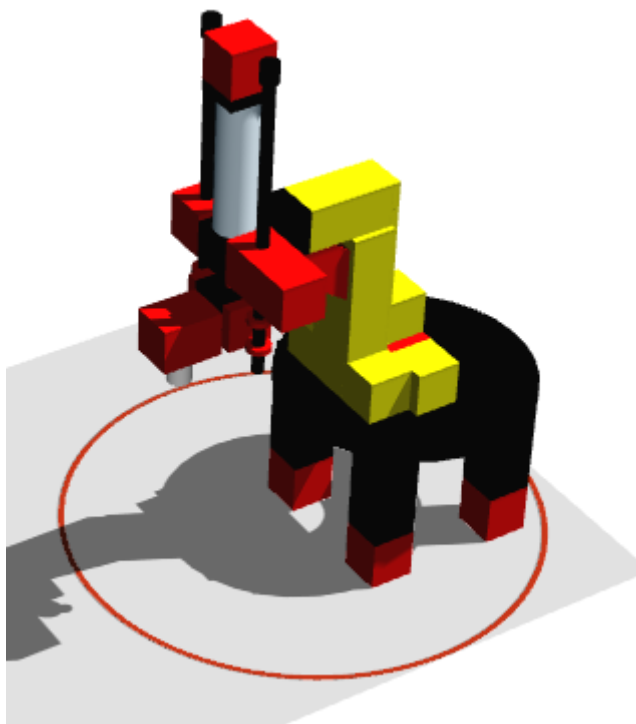
Obrázek 5-37: Nastavení animace dráhy kotouče pily



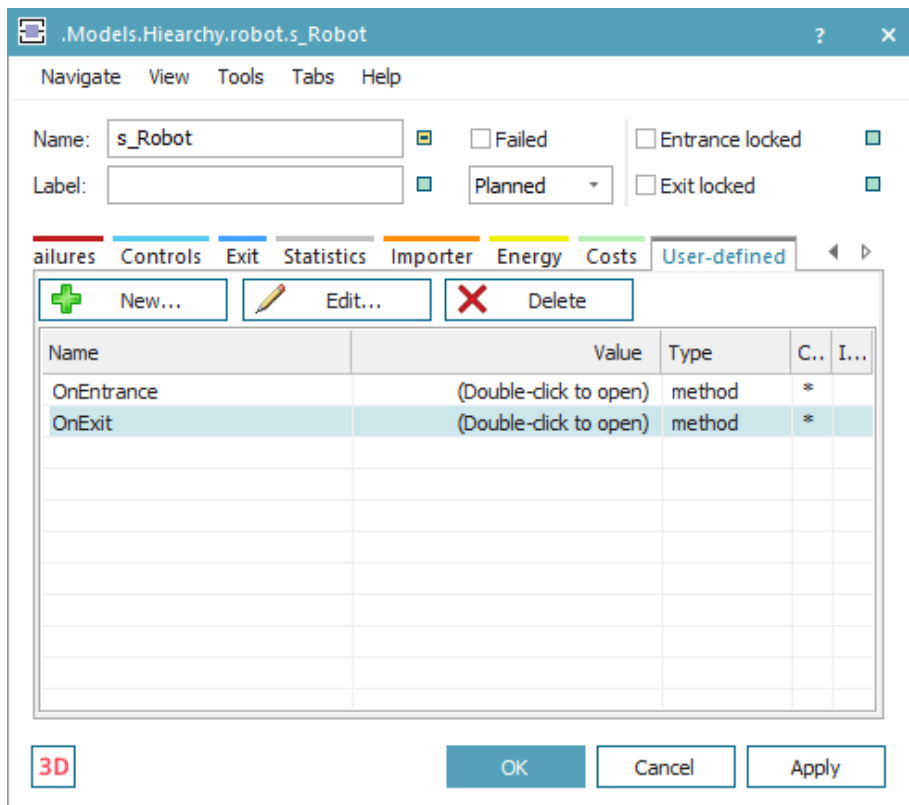
Obrázek 5-38: Ukázka použitých metod stroje Pila

## 5.5 Manipulační robot

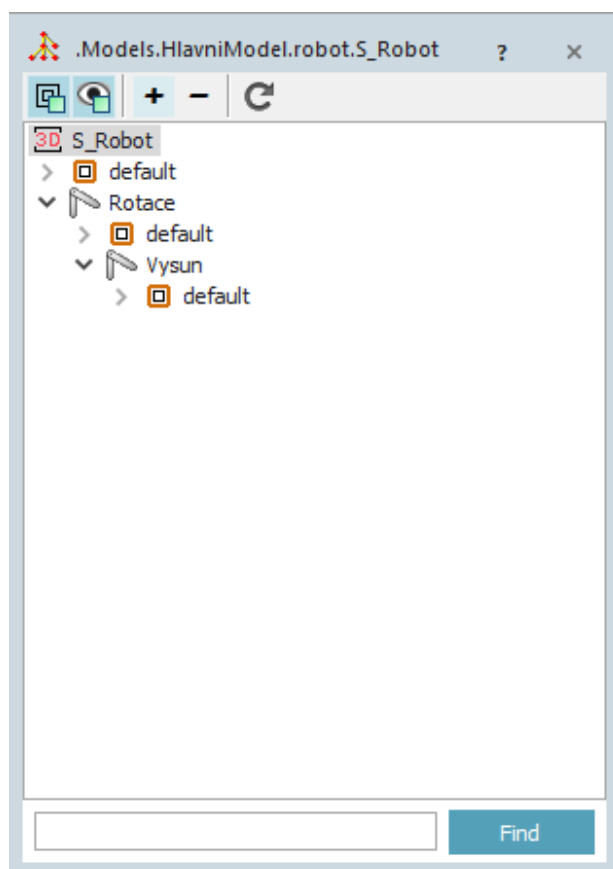
Manipulační robot má jedinou funkci, a to je manipulace výrobků z jednoho pásu na druhý. Robot lze vidět vymodelovaný na obrázku 5-39. Robot při vstupu rotuje hlavou nad určené místo u vstupního pásu. Poté mu hlava sjede dolů a přichytí si výrobek pomocí podtlaku. Animace trasy výrobku je zde vyřešena pomocí přichycení výrobku ke spodní části přísavky. Poté se samotná hlava zvedne a rotuje celá horní část až na určité místo nad druhým pásem. Zde se vysune hlava dolů a přísavka pustí výrobek na druhý pás. Toto vše nám provádí metoda `onEntrance`, která se spustí při vstupu výrobku do Manipulačního robota, což je opět dovoleno pouze hodnotou z řídicí aplikace. Druhá metoda, `onExit`, provádí po puštění výrobku na pás vrácení se do neutrální polohy stroje. Při programování animací musela být dodržena určitá struktura stroje, kterou lze vidět na obrázku 5-41. Nejdříve se vyberou části na rotaci a poté v této části se vyberou jednotlivé části hlavy pro posun nahoru a dolů. Bez nedodržení této struktury není možné vytvářet animaci.



Obrázek 5-39: Manipulační robot

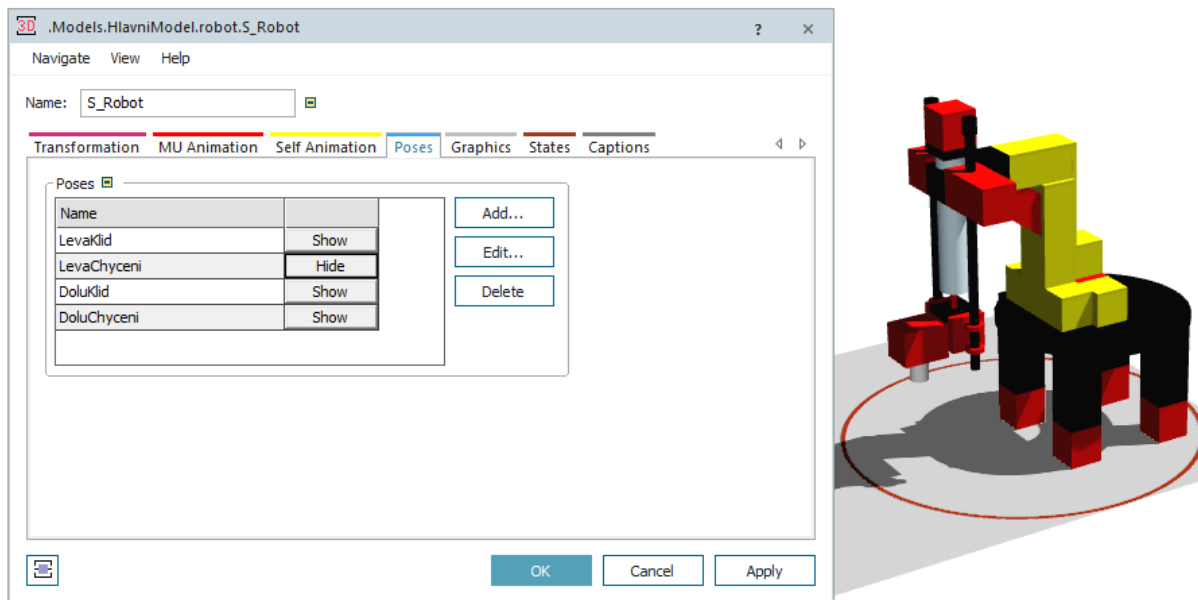


Obrázek 5-40: Ukázka použitých metod stroje Manipulační robot

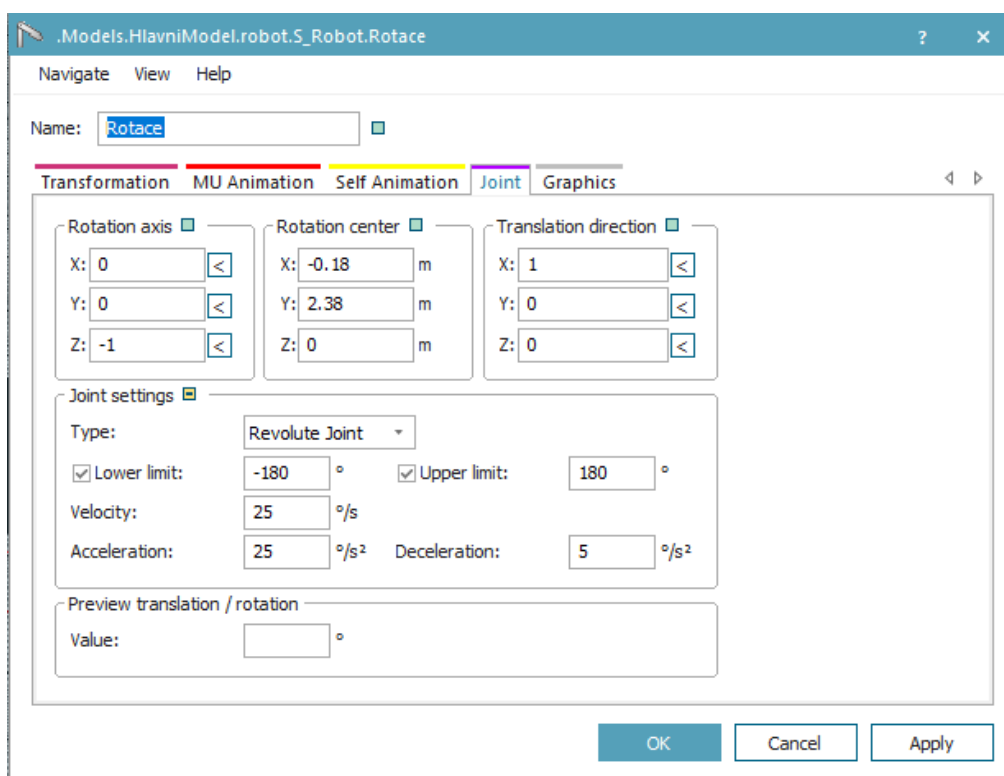


Obrázek 5-41: Grafická struktura modelu manipulačního robota

Na obrázku 5-42 můžeme vidět nastavení cílových poloh manipulačního robota. Tyto jednotlivé polohy se skládají z vytvořených pozic, viz. Obrázek 5-43. U těchto poloh se nastavuje buď to rotace nebo výsuv. Dále se nastavují maximální a minimální rozsahy, rychlost, zrychlování a zpomalování pohybu. Také samotná poloha osy rotace a její směr.



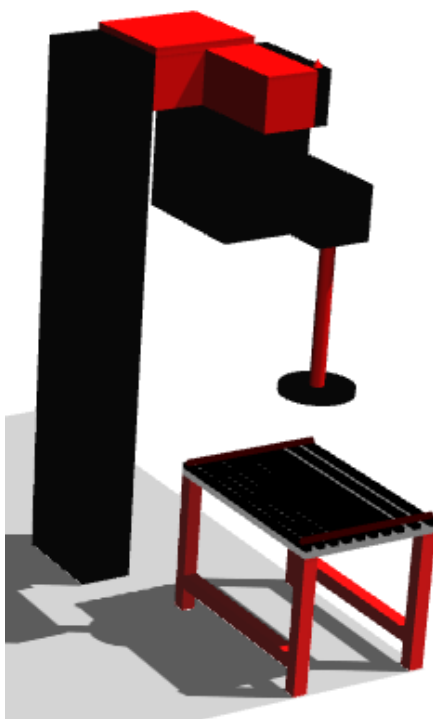
Obrázek 5-42: Ukázka polohy manipulačního robota



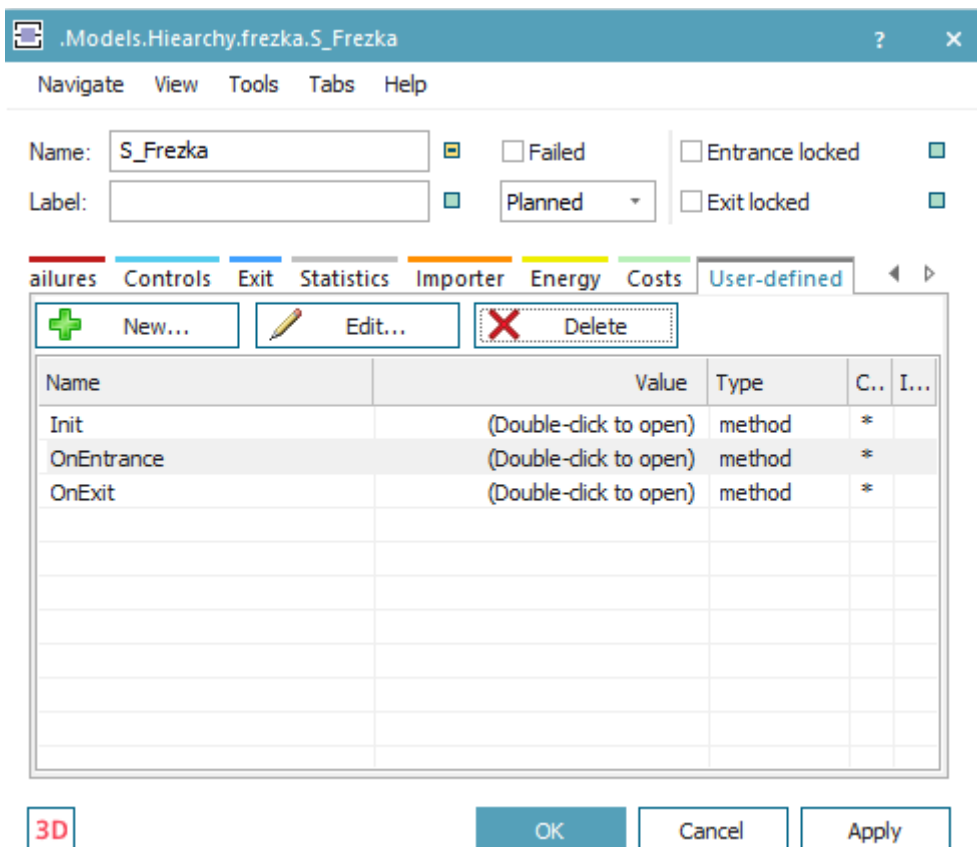
Obrázek 5-43: Ukázka nastavení polohy

## 5.6 Frézování

Frézka byla opět vymodelována pomocí základních geometrických útvarů z programu, viz. Obrázek 5-44, a je opět naprogramována pomocí metod, viz. obrázek 5-45. Opět zde funguje metoda Init, která po spuštění programu zavírá, jak vstup, tak výstup. Dále pokud příkáže řídicí aplikace pomocí hodnot určených pro frézku, otevře se vstup, dopočítá se procesní čas a provede se samostatný proces, to vše pomocí metody OnEntrance. Při tomto procesu je zde animace rotace frézy frézky. Ta je nastavena na délku procesu v metodě onEntrance. Animace probíhá rotací zmíněného kotouče kolem své osy. Dále probíhá zapisování procesních času do tabulky t\_Casy dle rozdělení jednotlivých barev. Zde je navíc event, když se rozbije frézka a tím se protáhne procesní čas o dobu opravy, tedy délku eventu Po přijmutí určité hodnoty z řídicí aplikace pro posun výrobku se výrobek posouvá dále po pásu a výstup se uzavírá pomocí metody OnExit.

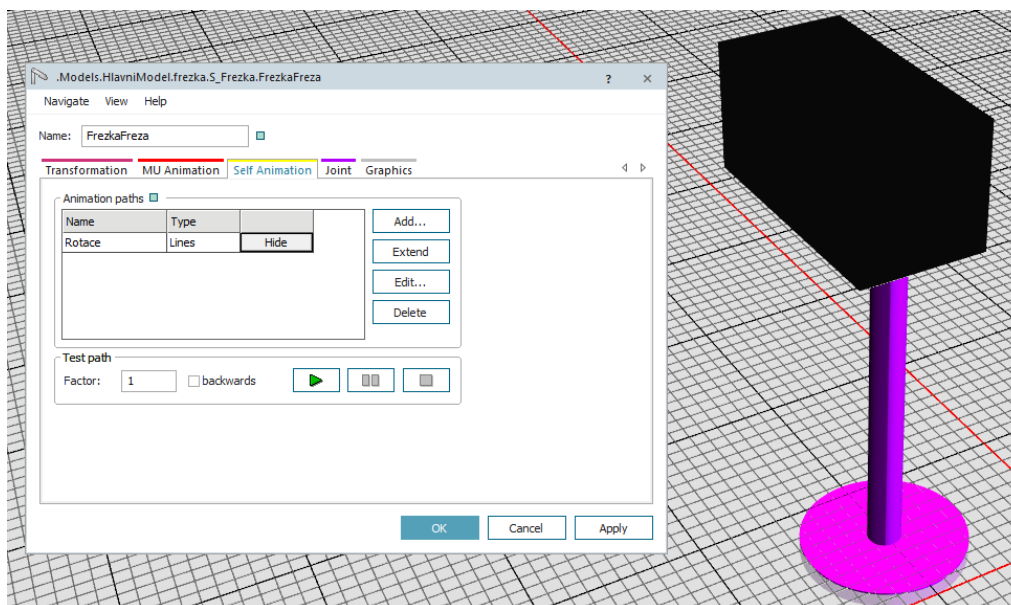


Obrázek 5-44: Vymodelovaná frézka



Obrázek 5-45: Ukázka použitých metod stroje Frézka

Na obrázku 5-46 je vidět opět samotné nastavení animace frézky, a to rotace.

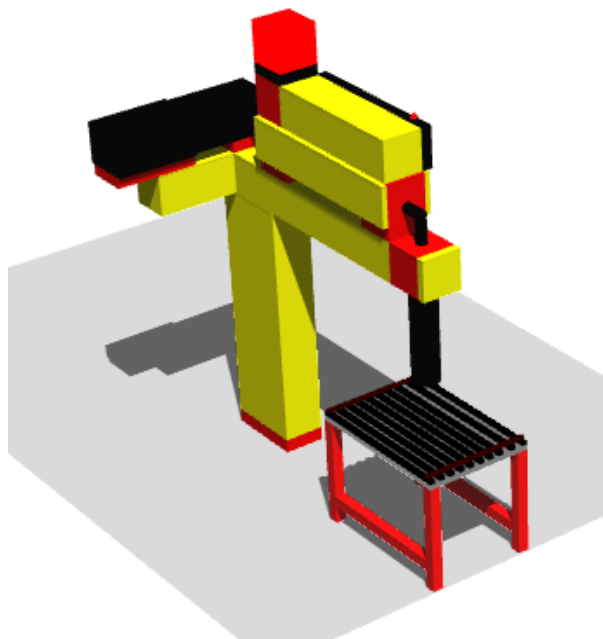


Obrázek 5-46: Animace frézy frézky

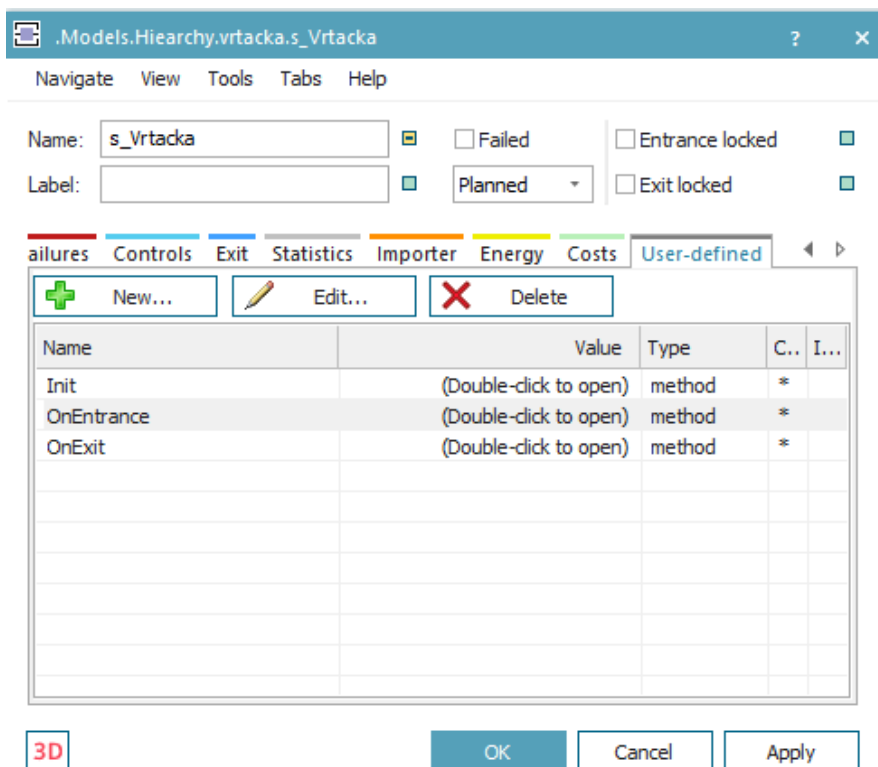
## 5.7 Vrtání

Vrtačka je opětovně vymodelována, viz. Obrázek 5-47, a naprogramována pomocí různých metod, viz. Obrázek 5-48, a to Init, OnEntrance a OnExit. Tyto metody zahrnují inicializaci

vstupu a výstupu při spuštění aplikace, otevření vstupu a dopočítání času procesu, provedením animace a následovány uzavřením výstupu po dokončení obrábění výrobku. Během procesu je vrták vrtačky animován rotací kolem své osy, když je výrobek opracováván na pracovišti. Dále dochází k zaznamenávání časových údajů do tabulky t\_Casy podle přidělených barev. V případě poruchy vrtačky se navíc vyskytuje událost, která prodlužuje celkový procesní čas o dobu trvání opravy, což představuje délku této události.

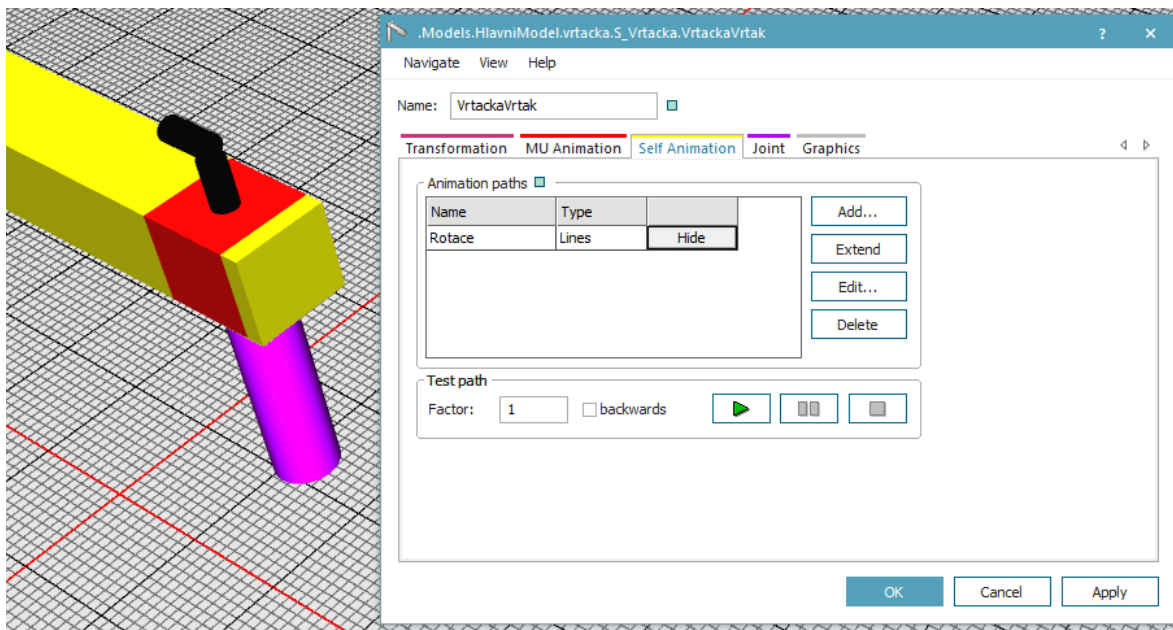


Obrázek 5-47: Vymodelovaná vrtačka



Obrázek 5-48: Ukázka použitých metod stroje Vrtačka



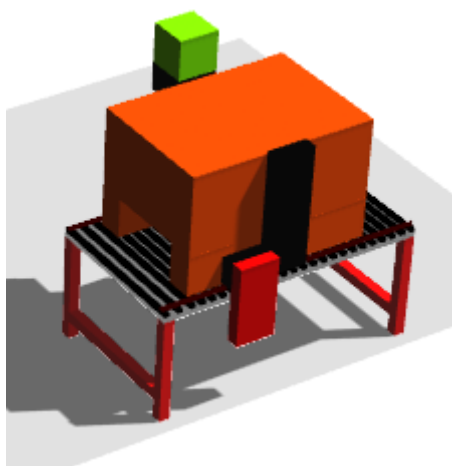


Obrázek 5-49: Animace vrtáku vrtačky

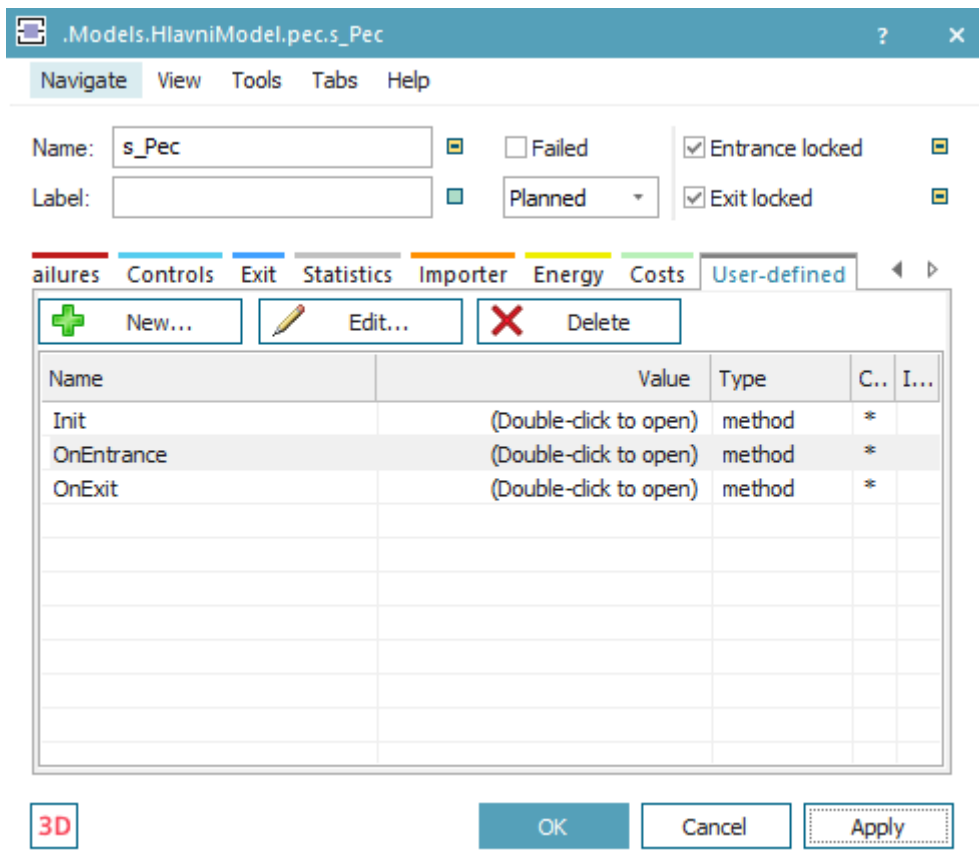
Na obrázku 5-49 je vidět opět samotné nastavení animace vrtáku, a to rotace.

## 5.8 Pec

Pec byla vymodelována, viz. Obrázek 5-50, a je naprogramována podobně jako předchozí stroje. Je opět naprogramována pomocí podobných metod, viz. obrázek 5-51, Init, OnEntrance a OnExit. Tyto metody zahrnují inicializaci vstupu a výstupu při spuštění aplikace, otevření vstupu a dopočítání času procesu a následovaný uzavřením výstupu po dokončení obrábění výrobku. Dále dochází k zaznamenávání časových údajů do tabulky t\_Casy podle přidělených barev. I zde byla potřeba nastavit, jako u všech ostatních strojů, dráha pro výrobek, kdy výrobek musí projíždět pecí.



Obrázek 5-50: Vymodelovaná pec



Obrázek 5-51: Ukázka použitých metod stroje Pec

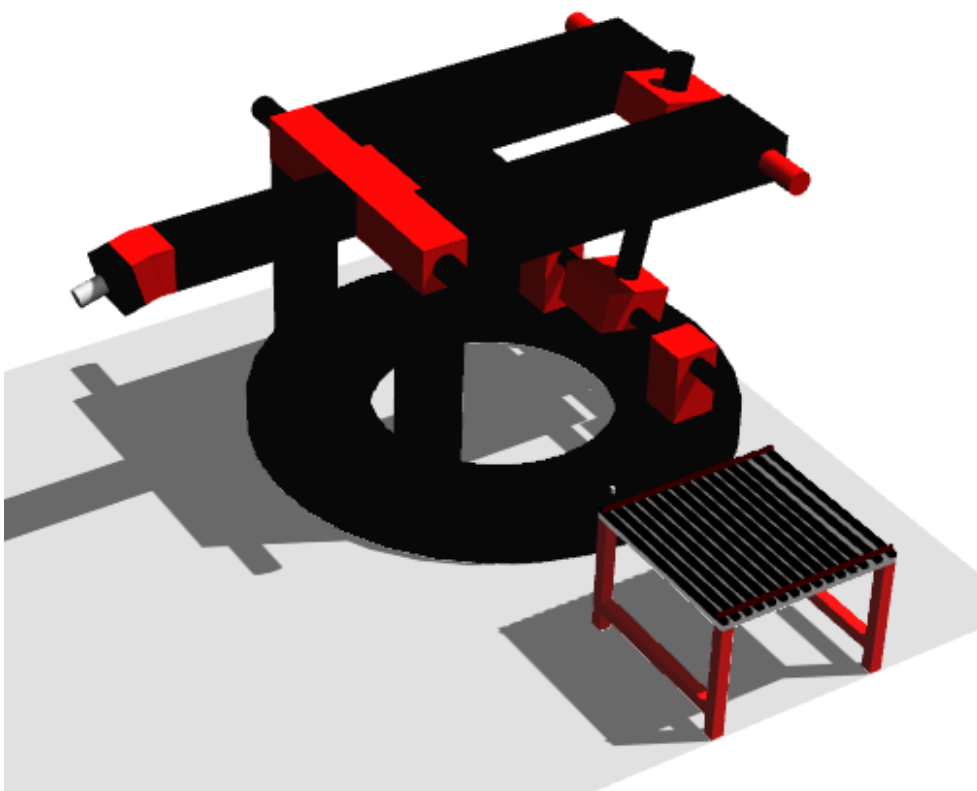
Na obrázku 5-52 je vidět ukázka kódu metody onEntrance. Jsou zde deklarované proměnné a čeká se, dokud nebude pec prázdná. Poté podle barvy se rozhodne, jaká část switchu se bude využívat pro dopočet času procesu. Jediná změna je v tom, kam se zapíše finální hodnota do tabulky t\_Casy.

```
param parameter : real, nextEventTime : real
waituntil self.~.empty = false
if self.~.empty = false
  var part : object
  var tab : table
  var i : integer
  part := self.~.cont
  part._3D.MaterialActive := true
  switch parameter
  case 1
    part.NextEventTime := nextEventTime;
    part.Name := "P_White"
    part.Barva := "Bila"
    self.~.ProcTime := part.NextEventTime - root.EventController.SimTime
    tab := t_Casy["Bila",1]
    tab[5,tab.YDIM + 1] := self.~.ProcTime
  case 2
    part.NextEventTime := nextEventTime;
    part.Name := "P_Red"
    part.Barva := "Cervena"
    self.~.ProcTime := part.NextEventTime - root.EventController.SimTime
    tab := t_Casy["Cervena",1]
    tab[5,tab.YDIM + 1] := self.~.ProcTime
  case 3
    part.NextEventTime := nextEventTime;
    part.Name := "P_Blue"
    part.Barva := "Modra"
    self.~.ProcTime := part.NextEventTime - root.EventController.SimTime
    tab := t_Casy["Modra",1]
    tab[5,tab.YDIM + 1] := self.~.ProcTime
  end
end
end
```

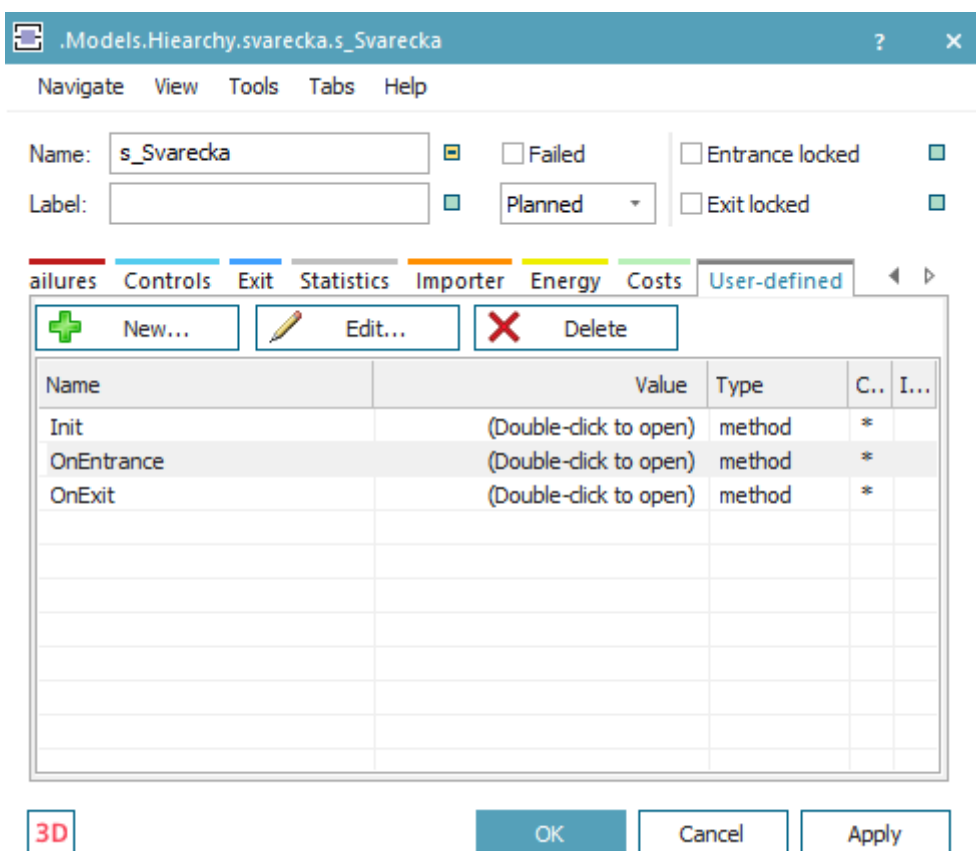
Obrázek 5-52: Ukázka kódu metody *OnEntrance*

## 5.9 Svařování

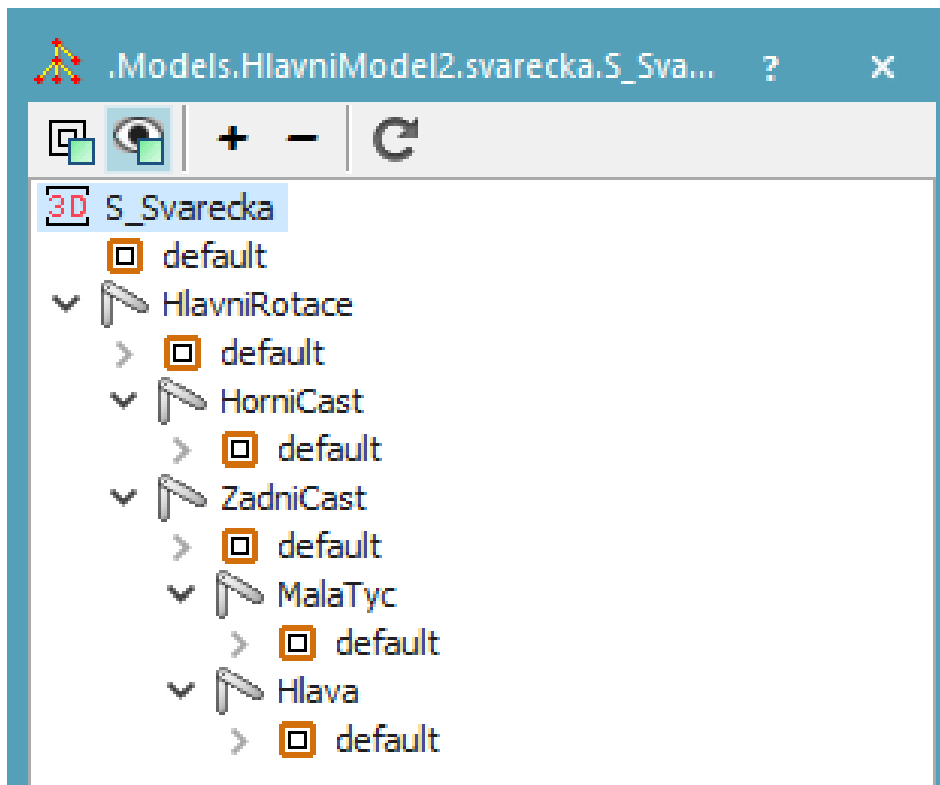
Svářečka byla opět vymodelována, viz. obrázek 5-53, a byly zde napsány opět 3 metody, *Init*, *onEntrance* a *onExit*, viz. obrázek 5-54. Svářečka je nejkomplexnější stroj celé linie, co se týká animací. Muselo se zde vytvořit 6 částí při modelování stroje, viz. obrázek 5-55, které se následovně použily pro samotné modelování 3 pracovních poloh, viz. obrázek 5-56. První poloha je ta, že je svářečka v klidu, ve své boční poloze. Druhá poloha je rotace hlavní celé části nad výrobní pás, kde bude čekat výrobek. Poslední částí je snížení polohy hlavy nad samotný výrobek, kde se simuluje proces sváření. Po dokončení procesu se hlava opět zvedne a samotný stroj rotuje do polohy klidu. Také celý stroj je ovládán pomocí metod a časy procesů se zapisují do tabulky.



Obrázek 5-53: Vymodelovaná svářečka

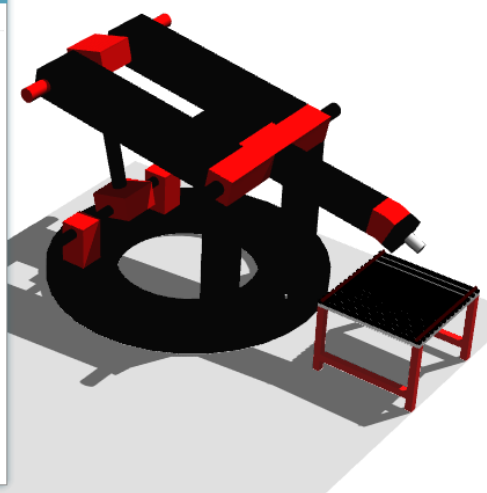
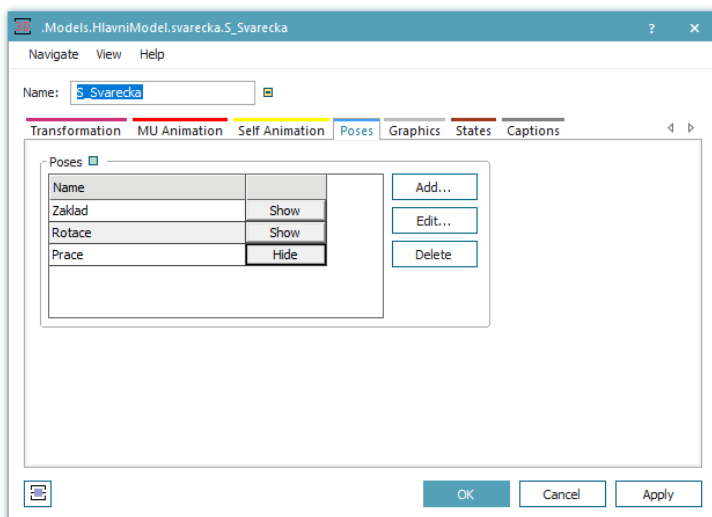


Obrázek 5-54: Ukázka použitých metod stroje Svářečka

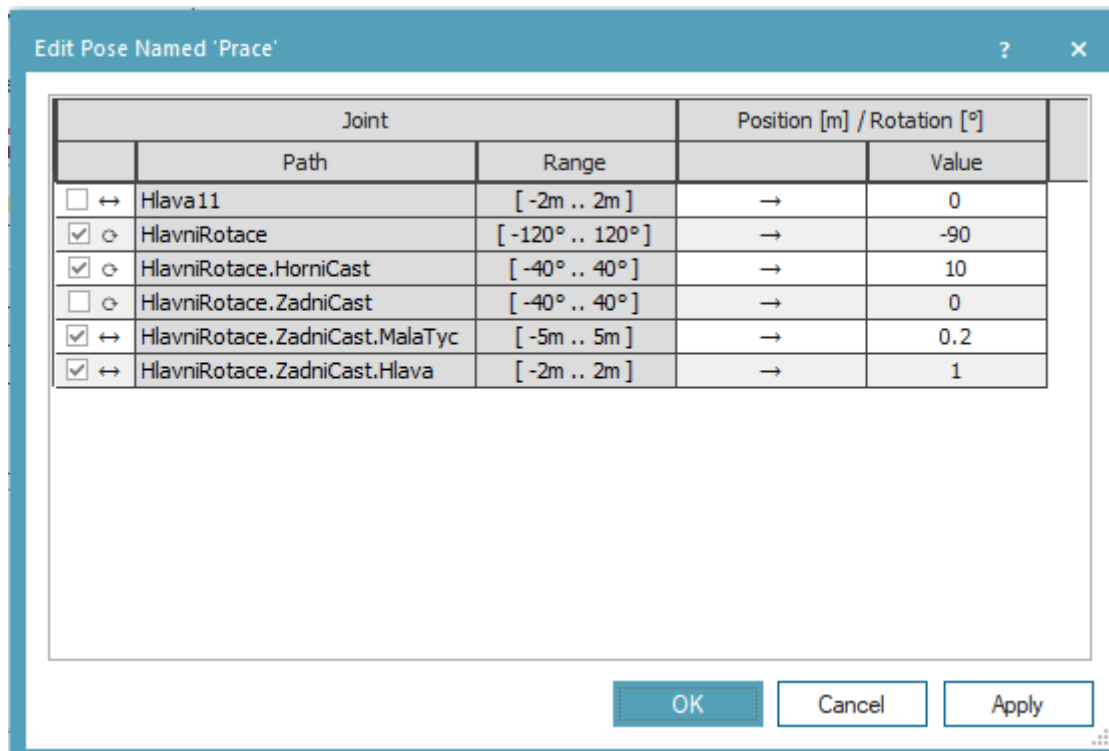


Obrázek 5-55: Grafická struktura modelu svářečky

Na obrázku 5-55 je vidět grafická struktura modelu svářečky. Jako hlavní část slouží HlavniRotace. Pod touto částí byla potřeba vymodelovat HorniCast a ZadniCast pro vytvoření požadované polohy, která slouží k animacím. Poté pod částí ZadniCast byly vymodelovány části MalaTyc a Hlava. Tato struktura poté sloužila k modelaci jednotlivých poloh, které jsou vidět na obrázku 5-56. Detail polohy Práce je vidět na obrázku 5-57, kde se zapínají jednotlivé polohy a k nim se určuje rozsah.



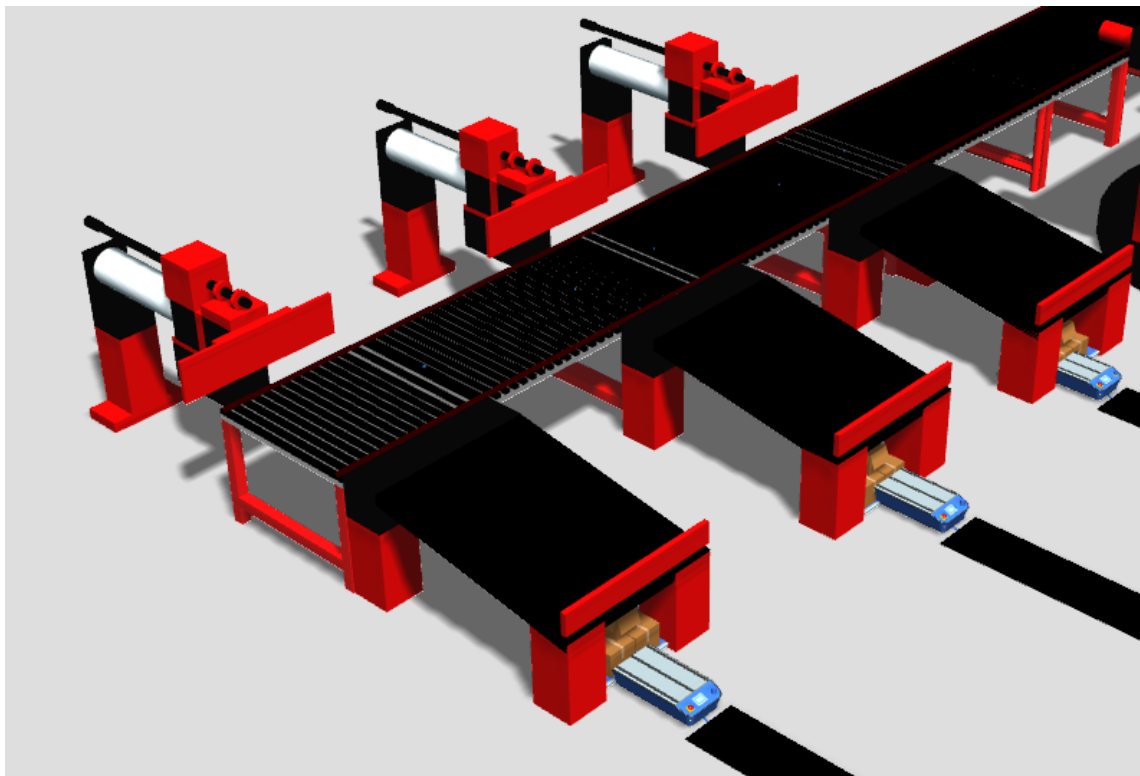
Obrázek 5-56: Ukázka pracovní polohy svářečky



Obrázek 5-57: Ukázka parametrů polohy práce

## 5.10 Třídění a transport výrobků

Tato část již neexistuje v realitě, je to doplnění stávajícího reálného modelu, jak by se mohl do budoucna model rozšířit. Po stanovišti svářečka jede puk po pásu. Dle barvy se jednotlivé puky roztrídí do přiřazených bufferů pomocí bočního panelu, který by je mohl posouvat. Puk by sjel po pásu dolů, kde by na něj čekalo přepravní vozidlo. To by pomocí předem vytvořené cesty, pomocí magnetických pásek, přepravilo výrobek zpět do počátečního skladu a výrobek zde vykloupilo pomocí zvedací plošiny na připravené místo. Tato část simulace není řízená pomocí metod, ale jen nastavení. Pokud přijede na pás výrobek, je dle barvy roztríděn a pokud je v bufferu na konci pásů výrobek, přepravní vozík ho automaticky převezde zpět do skladu na začátku. Tyto puky ale nevstupují znovu do výrobního procesu.

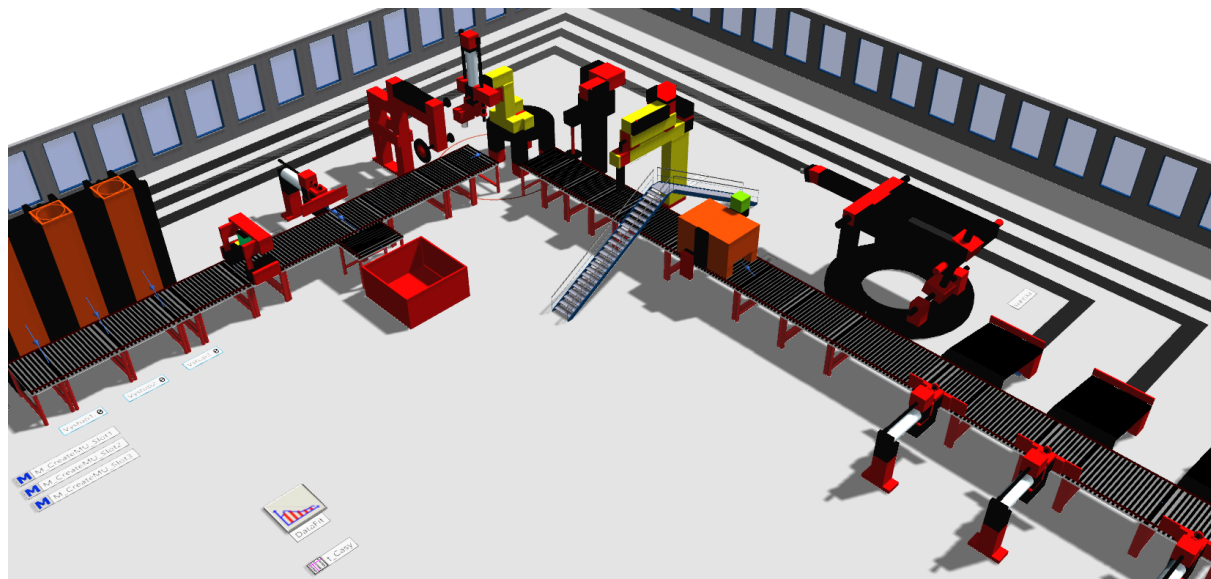


Obrázek 5-58: Vymodelovaná potencionální třídička



Obrázek 5-59: Ukázka kódu ovládní vozíku a vozíku

Na obrázku 5-60 lze vidět celkový pohled na model simulace po vymodelování všech částí reálného modelu a části Třídění a transport výrobků, která v reálu neexistuje.

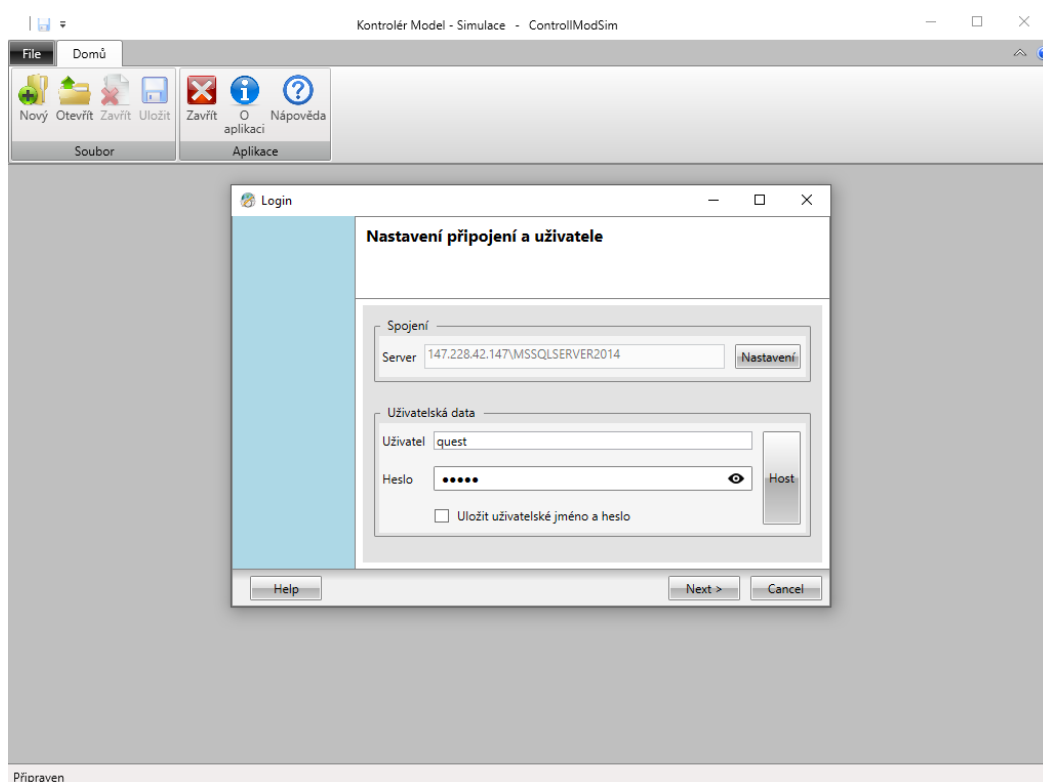


Obrázek 5-60: Celkový pohled na model simulace

### 5.11 Propojení s modelem

Po zapnutí řídicí aplikace je nutné se přihlásit jako host, jako to lze vidět na obrázku 5-61. Dále v aplikaci lze vidět vysvětlivku k jednotlivým událostem, a to jak v českém jazyce, tak v anglickém, viz. Obrázek 5-62. Dalším krokem řídicí byla potřeba nejdříve nastavit cestu k simulačnímu modelu, co lze vidět na obrázku 5-63. Dále nastavit cestu na hlavní frame, kde se nachází vše vymodelované. Poté v aplikaci jsou jednotlivé potřebné eventy, ale byla potřeba nastavit cestu ke správným stanicím, kde se mají použít. Tedy když se má otevřít např. rentgenová kontrola a začít proces kontroly, bylo potřeba nastavit pro hodnotu řídicí aplikace RentgenovaKontrolaStart cestu na stroj rentgenová kontrola. Toto nastavení bylo uloženo a použito pro všechny hodnoty řídicí aplikace.





Obrázek 5-61: Spuštění aplikace

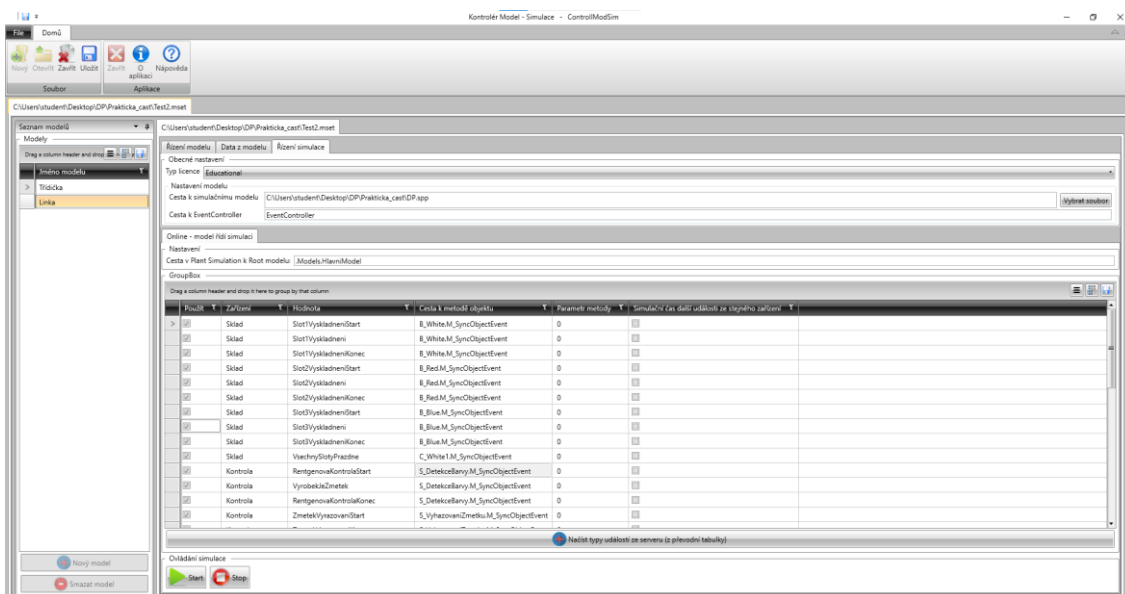
Nastavení Report

**Převodní tabulka z Arduino**

Drag a column header and drop it here to group by that column

ModelDevice	Index	Ukládat	Klíč	Hodnota	Určit proces. čas k události	Popis - význam	Popis - význam anglicky
>	1	<input checked="" type="checkbox"/>	10	Slot1VyskladneniStart		Začátek operace vyskladnění materiálu ze skladu, s	Start of the operation to unload material from the v
Sklad	2	<input checked="" type="checkbox"/>	11	Slot1Vyskladneni		Vyskladnění materiálu ze skladu, slotu č. 1 na dopra	Unloading material from the warehouse, slot 1 ontc
Sklad	3	<input checked="" type="checkbox"/>	12	Slot1VyskladneniKonec		Ukončení operace vyskladnění materiálu ze skladu,	Completion of the material unloading operation fro
Sklad	4	<input checked="" type="checkbox"/>	13	Slot2VyskladneniStart		Začátek operace vyskladnění materiálu ze skladu, s	Start of the material unloading operation from the \
Sklad	5	<input checked="" type="checkbox"/>	14	Slot2Vyskladneni		Vyskladnění materiálu ze skladu, slotu č. 2 na dopra	Unloading of material from the warehouse, slot 2 or
Sklad	6	<input checked="" type="checkbox"/>	15	Slot2VyskladneniKonec		Ukončení operace vyskladnění materiálu ze skladu	Completion of the operation to unload the material
Sklad	7	<input checked="" type="checkbox"/>	16	Slot3VyskladneniStart		Začátek operace vyskladnění materiálu ze skladu, s	Start of the material unloading operation from the \
Sklad	8	<input checked="" type="checkbox"/>	17	Slot3Vyskladneni		Vyskladnění materiálu ze skladu, slotu č. 3 na dopra	Unloading of material from the warehouse, slot 3 or
Sklad	9	<input checked="" type="checkbox"/>	18	Slot3VyskladneniKonec		Ukončení operace vyskladnění materiálu ze skladu,	Completion of the material unloading operation fro
Sklad	10	<input checked="" type="checkbox"/>	19	VsechnySlotyPrazdne		Informace, že všechny sloty skladu jsou prázdné.	Information that all storage slots are empty.
Kontrola	11	<input checked="" type="checkbox"/>	30	RentgenovaKontrolaStart		Zahájení rentgenové kontroly obrobku.	Starting the X-ray inspection of the workpiece.
Kontrola	12	<input checked="" type="checkbox"/>	31	VyrobekJeZmetek		Obrobek je vyhodnocený jako zmetek	The workpiece is evaluated as a junk
Kontrola	13	<input checked="" type="checkbox"/>	4	RentgenovaKontrolaKonec	30	Ukončení rentgenové kontroly výrobku.	Completion of X-ray inspection of the product.
Kontrola	14	<input checked="" type="checkbox"/>	32	ZmetekVyrzovaniStart		Zahájení fyzického vyřazení obrobku ze systému, pr	Initiate physical removal of the workpiece from the
Kontrola	15	<input checked="" type="checkbox"/>	33	ZmetekVyrzovaniKonec		Ukončení fyzického vyřazení obrobku ze systému, p	Terminate physical removal of workpiece from syste
Kontrola	16	<input checked="" type="checkbox"/>	34	RezaniStart	31	Zahájení operace řezání.	Start cutting operation.
Kontrola	17	<input checked="" type="checkbox"/>	35	RezaniKonec		Ukončení operace řezání.	Termination of the cuttino operation.

Obrázek 5-62: Převodní tabulka z Arduino



Obrázek 5-63: Nastavení řídicí aplikace

## 6. Simulace dlouhodobého chování

Pro přepnutí modelu do simulace dlouhodobého chování jsou potřebná vstupní data. Tyto data byla získána z tabulky `t_Casy`, kam se ukládají všechny časy ze stanovišť strojů, které byly popsány v předešlé kapitole.

Po spuštění simulace řízené eventy aplikací bylo tedy sesbíráno celkem 93 záznamů přes všechny stanoviště strojů barvy puků z tabulky `t_Casy`. Tabulka je rozdělena do tří hlavních sloupců, které symbolizují barvy puků. Řádky symbolizují jednotlivé stroje až na první řádek, ten symbolizuje vstupy jednotlivých puků s časovým razítkem. Tabulka je naformátována na tabulky, které jsou dále naformátovány již na časy. Tato tabulka sloužila jako vstup pro další analýzu, viz. Tabulka 6-1.

Tabulka 6-1: Tabulka záznamů časů `t_Casy`

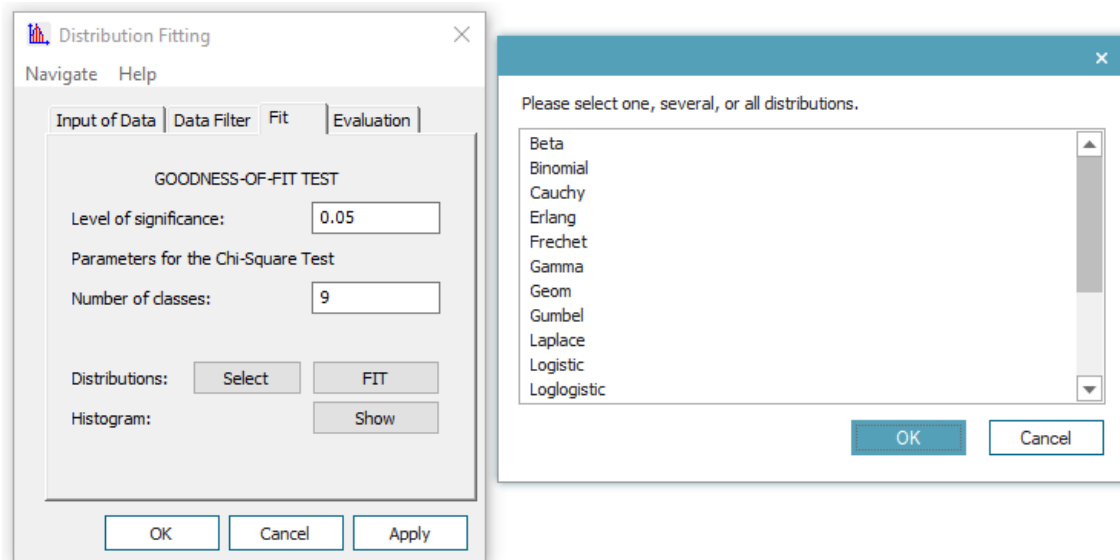
	string 0	table 1	table 2	table 3
string		Bila	Cervena	Modra
1	Vstup	Vstup_Bila	Vstup_Cervena	Vstup_Modra
2	RentgenováKontrola	RentKontrola_Bila	RentKontrola_Cervena	RentKontrola_Modra
3	Pila	Pila_Bila	Pila_Cervena	Pila_Modra
4	Frezka	Frezka_Bila	Frezka_Cervena	Frezka_Modra
5	Vrtacka	Vrtacka_Bila	Vrtacka_Cervena	Vrtacka_Modra
6	Pec	Pec_Bila	Pec_Cervena	Pec_Modra
7	Svarecka	Svarecka_Bila	Svarecka_Cervena	Svarecka_Modra

Na obrázku 6-64 lze vidět příklad části uložených časů tabulky 6-1.

	time 1
1	2.3386
2	2.2997
3	2.2869
4	2.2620
5	2.2566
6	2.2531
7	2.2704
8	2.2211
9	2.2107
10	2.2060
11	2.2817
12	2.2762
13	2.2809
14	2.2633
15	2.1911
16	2.3452
17	2.3192
18	2.3149
19	2.2938
20	2.3198
21	2.2814
22	2.2983
23	2.2758
24	2.2808

Obrázek 6-64: Ukázka sběru časů pro jednu barvu puku určitého stroje

Právě na tyto sesbíraná data, která byla uložena v tabulce 6-1, byla použita statistická funkce DataFit, která je součástí programu. Do této funkce se ručně nahrály jednotlivá data z tabulky 6-1, tedy celkem 21 sloupců, a bylo možné vybrat z více než dvaceti distribučních funkcí, kde program vyzkoušel každou možnou distribuční funkci, jak lze vidět na obrázku 6-65.



Obrázek 6-65: Použití DataFitu – výběr distribučních funkcí

Po vložení získaných dat bylo možné provést evaluaci. Získaný report nabízí několik doporučených distribučních funkcí. Tyto doporučení vznikly na základě proběhnutých tří testů, jak je vidět na obrázku 6-66, a to Chi-Square, Kolmogorov-Smirnov a Anderson-Darling. Bohužel u některých distribučních funkcí program nebyl schopen vyhodnotit test pozitivně a vrátit výsledek.

#### Chi-Square

Distribution	Chi statistic	Chi value	Result Chi
Normal	8.7603	12.5875	true
Uniform	8.7912	12.5875	true
Logistic	9.5824	12.5875	true
Weibull	13.7363	12.5875	false
Paralogistic	16.1099	12.5875	false
Triangle	17.4945	11.0650	false
Gamma	27.7057	12.5875	false
Erlang	31.6812	11.0650	false
Loglogistic	35.8901	12.5875	false
Cauchy	46.4309	12.5875	false
Negexp	56.0659	14.0641	false
Lognorm	61.3508	12.5875	false
Pareto	135.1868	12.5875	false
Gumbel	1000.0000		

#### Kolmogorov-Smirnov

Distribution	KS statistic	KS value	Result KS
Normal	0.8709	1.3580	true
Uniform	1.0539	1.3580	true
Logistic	1.0646	1.3580	true
Weibull	1.0688	1.3580	true
Paralogistic	1.3271	1.3580	true
Triangle	1.2249	1.3580	true
Gamma	1.2103	1.3580	true
Erlang	2.4580	1.3580	false
Loglogistic	1.7982	1.3580	false
Cauchy	1.4232	1.3580	false
Negexp	2.2723	1.3580	false
Lognorm	1.5833	1.3580	false
Pareto	3.0347	1.3580	false
Gumbel	1.3696	1.3580	false

#### Anderson-Darling

Distribution	AD statistic	AD value	Result AD
Normal	0.8163	2.4920	true
Uniform	3.5024	2.4920	false
Logistic	1.4671	2.4920	true
Weibull	1.6346	2.4920	true
Paralogistic	3.1777	2.4920	false
Triangle	6.7183	2.4920	false
Gamma	2.8323	2.4920	false
Erlang	11.9233	2.4920	false
Loglogistic	8.3917	2.4920	false
Cauchy	2.6792	2.4920	false
Negexp	9.5649	2.4920	false
Lognorm	5.3452	2.4920	false
Pareto	26.6037	2.4920	false
Gumbel	3.3394	2.4920	false

Obrázek 6-66: Ukázka doporučení testů

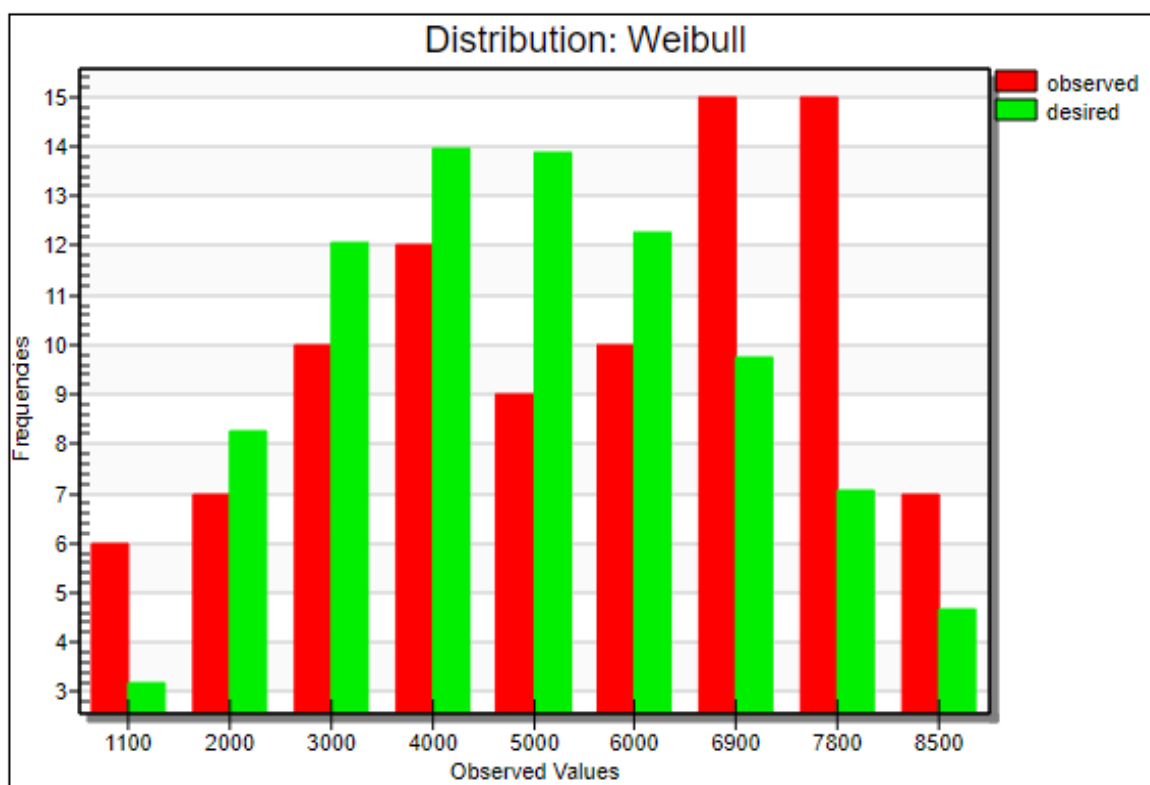
Pro distribuční funkce program připravil i jejich parametry, jak by se s nimi mělo pracovat, jak je vidět na obrázku 6-67.

*Estimated parameters*

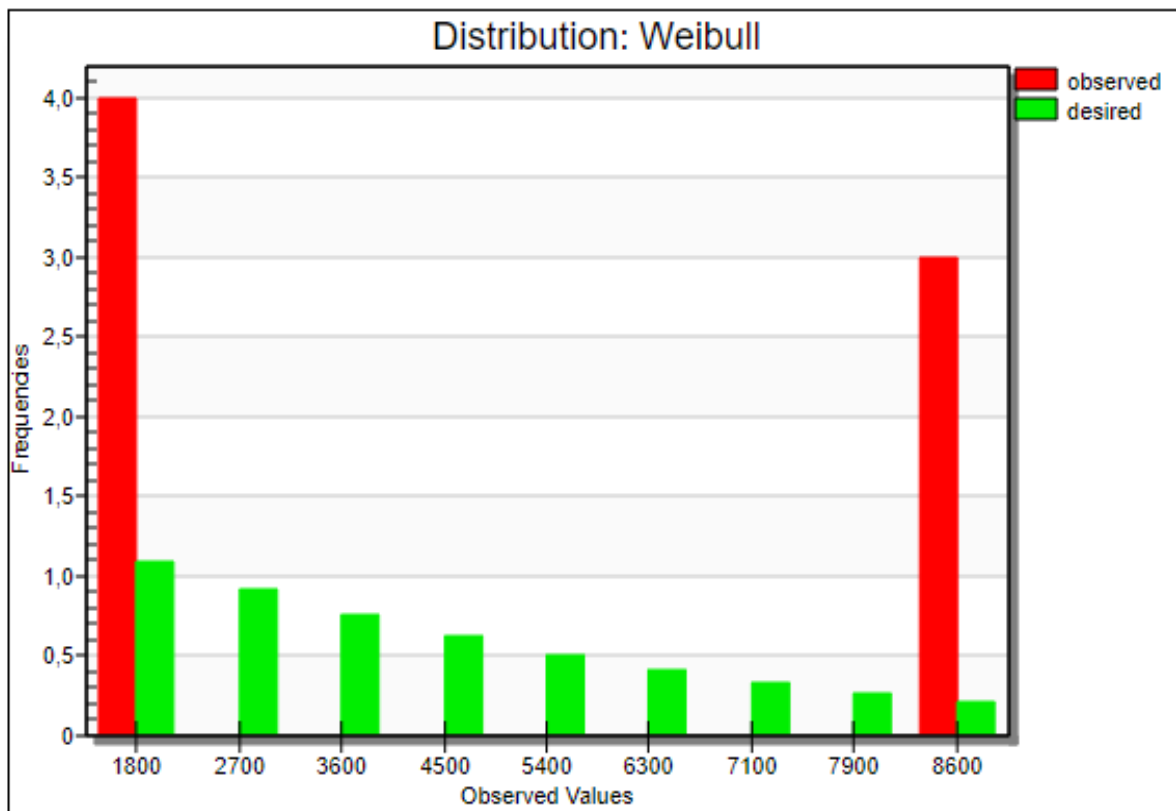
Distribution	Parameter 1	Parameter 2	Parameter 3	SimTalk
Cauchy	Mu = 5.3975253	Theta = 0.00615647500000005		var r: real := z_cauchy(1, 5.3975253, 0.00615647500000005)
Loglogistic	Alpha = 5.39454556648183	Beta = 430.040563028002		var r: real := z_loglogistic(1, 5.39454556648183, 430.040563028002)
Logistic	Mu = 5.39459354946237	Sigma = 0.0227532405382017		var r: real := z_logistic(1, 5.39459354946237, 0.0227532405382017)
Normal	Mu = 5.39459354946237	Sigma = 0.0226305806659861		var r: real := z_normal(1, 5.39459354946237, 0.0226305806659861)
Pareto	Alpha = 170.741252552992	Theta = 5.36316079040393		var r: real := z_pareto(1, 170.741252552992, 5.36316079040393)
Paralogistic	Alpha = 169.99999995094	Theta = 5.57870920816618		var r: real := z_paralogistic(1, 169.99999995094, 5.57870920816618)
Triangle	c = 5.39459354946237	a = 5.3315133	b = 5.5488765	var r: real := z_triangle(1, 5.39459354946237, 5.3315133, 5.5488765)
Uniform	Start = 5.3315133	Stop = 5.5488765		var r: real := z_uniform(1, 5.3315133, 5.5488765)
Negexp	Beta = 5.39459354946237			var r: real := z_negexp(1, 5.39459354946237)
Gumbel	Mu = 5.39459354946237	Sigma = 0.0227532405382017		var r: real := z_gumbel(1, 5.39459354946237, 0.0227532405382017)

Obrázek 6-67: Ukázka z výsledků DataFit

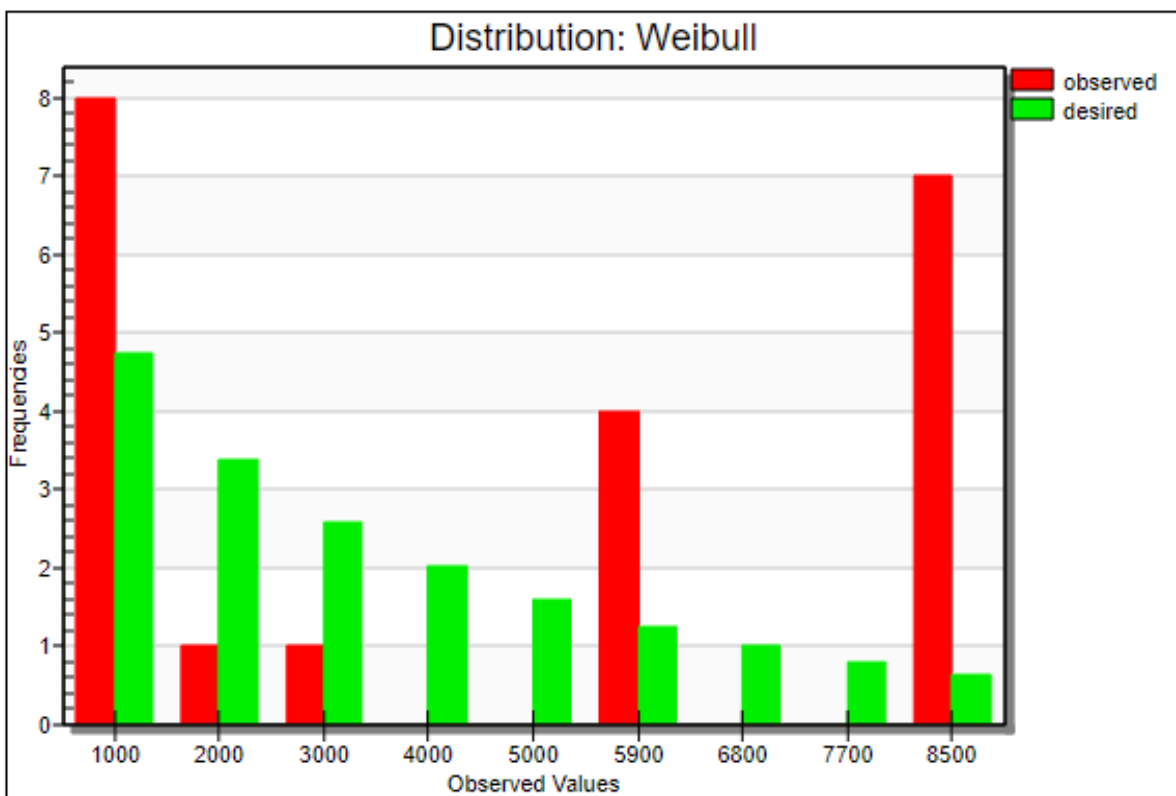
Dalším výstupem funkce DataFit je grafické porovnání sledovaných, červený sloupec, a navrhovaných dat, zelený sloupec, jak je vidět na obrázku 6-68. Tímto stylem se provedlo 5 testů pro celou výrobní linii, a to pro distribuční funkce: normální rozdělení, exponenciální, triangulární, Weibullovo a gamma pro každou barvu puku. Na obrázcích 6-68, 6-69 a 6-70 je uveden příklad grafického vyhodnocení vstupů do produkční linie pro puky bílé, červené a modré.



Obrázek 6-68: Ukázka z grafických výsledků DataFit 1

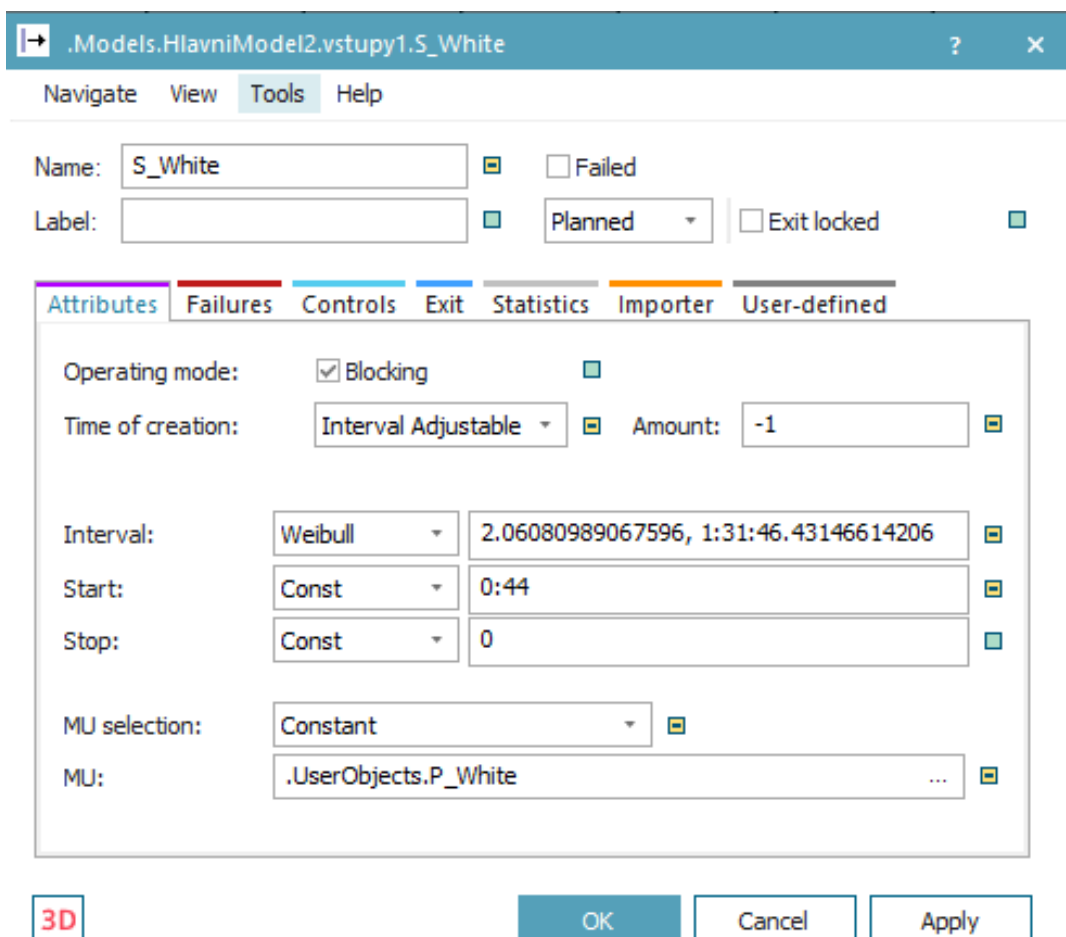


Obrázek 6-69: Ukázka z grafických výsledků DataFit 2

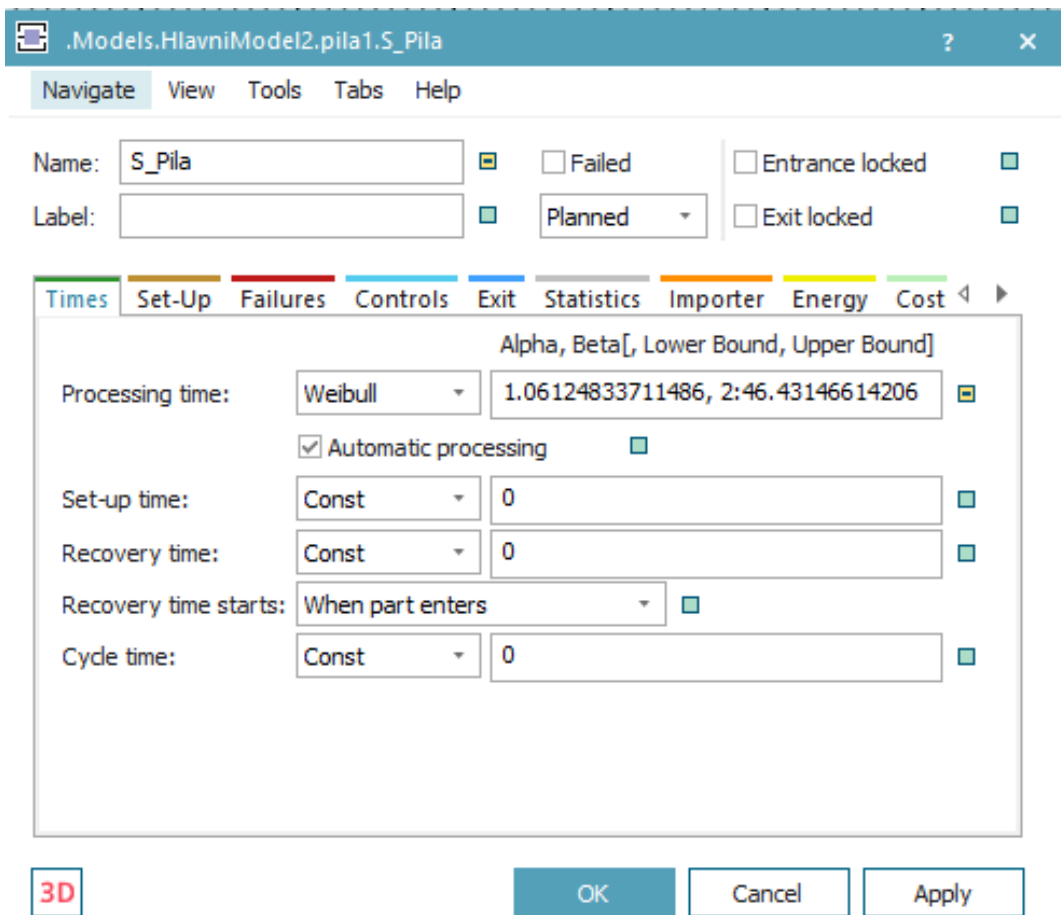


Obrázek 6-70: Ukázka z grafických výsledků DataFit 3

Dalším krokem bylo duplikování původního hlavního framu. Zde byla potřeba nastavit jednotlivá pracoviště. Nastavení jednotlivých pracovišť souviselo s nastavením časových hodnot a metod. U vstupů se nastavoval interval generování výrobků, jak lze vidět na obrázku 6-71. Zde lze vidět použité Weibullovo rozdělení. U strojů se naopak nastavovala délka procesů obrábění, což lze vidět na obrázku 6-72. U metod se musela vymazat část, která souvisela s řízením pomocí hodnot z řídicí aplikace. Zapisování časových hodnot do tabulky t\_Casy a logická rozhodování, např. u Vyhazovače zmetků, zůstala stejná. Samotné nastavení dlouhodobé simulace bylo provedeno na dvě stě čtyřicet hodin.



Obrázek 6-71: Ukázka nastavení vstupů pro dlouhodobou simulaci



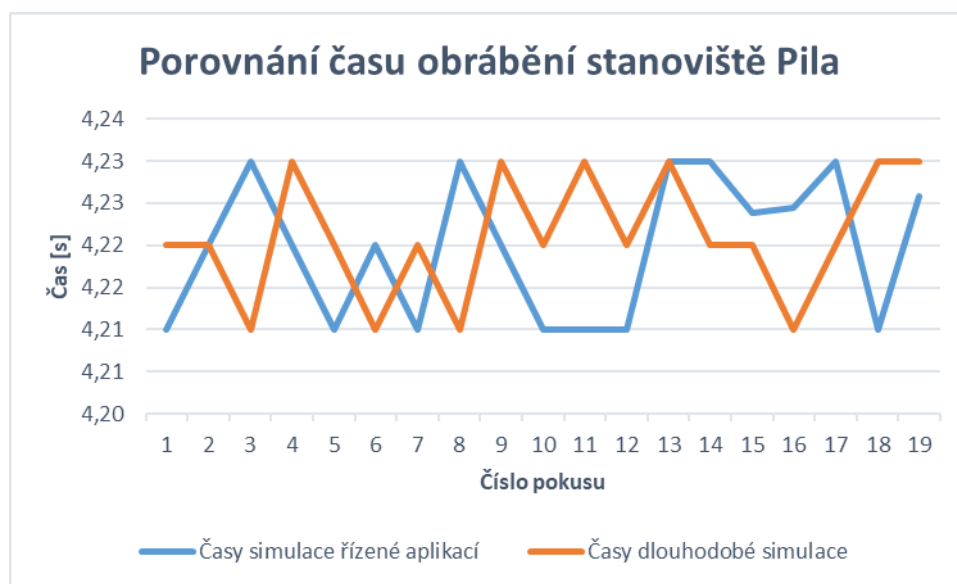
Obrázek 6-72: Ukázka nastavení pily pro dlouhodobou simulaci



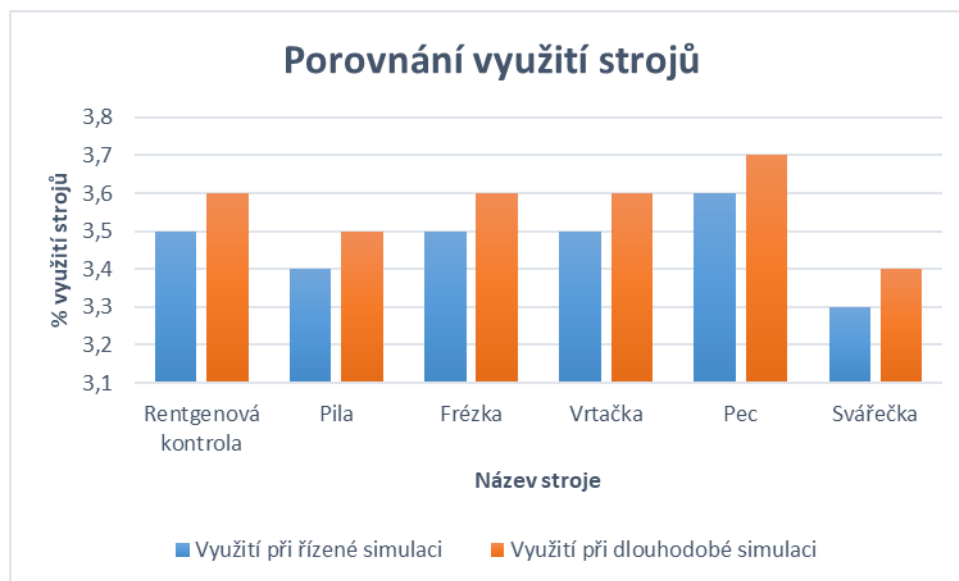
## 7. Analýza výsledků

Jako první byl proveden experiment na ověření správnosti naprogramování simulace. Tedy simulace řízená řídicí aplikací se zapnula a běžela vedle reálného modelu, který byl zapnutý. Ze simulovaného modelu se sesbírala data o vygenerovaných kusech a na reálném modelu se spočítaly puky, které prošly reálným modelem výrobní linie. Tento experiment proběhl úspěšně, protože na konci obou simulací, jak reálné, tak počítačové, byl napočítán stejný počet puků v obou experimentech. Tímto bylo docíleno ověření simulačního modelu.

Po ověření simulačního modelu se přistoupilo pro experimenty dlouhodobých simulací, které byly určeny na dvě stě čtyřicet hodin. Po provedení pěti dlouhodobých simulací pro vybrané distribuční funkce bylo zjištěno, že jednotlivé délky procesů na strojích se liší v minimální hodnotě. Toto porovnání bylo vyhodnoceno v excelu a lze vidět příklad vyhodnocení stanoviště Pila na obrázku 7-73. Je zde porovnáváno stejný počet cyklů obrábění na stroji pro stejnou barvu puku. Je to určeno velice podobnou délkou jednotlivých výrobních procesů, které se nelišily natolik u jednotlivých barvách puků na jednotlivých strojích. Dále při testech se vyhodnocovalo využití strojů, které lze vidět na obrázku 7-74. Celkově velice nízké využití je způsobeno tím, že při simulaci řízené řídicí aplikací byl vždy pouze jeden výrobek ve výrobní línii. Můžeme ale vidět na obrázku 7-74, že při dlouhodobé simulaci bylo navýšeno využití strojů, a to o necelé 2%.



Obrázek 7-73: Porovnání času obrábění stanoviště Pila



Obrázek 7-74: Porovnání využití strojů

Naopak velice zajímavý výsledek byl při simulaci vstupů puků do oběhu. Původní simulace, která byla řízena aplikací, měla omezenou délku a celkem trvala dvě hodiny a třicet jedna minut. Prodlevy mezi vstupy jednotlivých puků dle barvy byly jak malé, tak i větší. Bylo to dáno náhodným vkládáním puků do reálného modelu. Po dvě stě čtyřiceti hodinách dlouhodobých simulací pro jednotlivé distribuční funkce byla sestavena Tabulka 7-2. Ta nám říká, kolik kusů puků bylo vygenerováno pro jednotlivé barvy pro jednotlivé distribuční funkce.

Tabulka 7-2: Výsledek simulace pro vstupy v kusech

	P_White	P_Red	P_Blue
Normal	negativni hodnoty		
Ngexp	2352	2004	2020
Triangel	1739	2032	2029
Weibull	1794	1986	1981
Gamma	1802	2028	1976

Pro normální rozdělení program byl schopen určit parametry, ale hodnoty se mohly dostat do záporných čísel, a proto simulace neproběhla. Zde program lehce selhal, že po analýze neposkytl dolní a horní mez. Exponenciální rozdělení šlo od první stanice P\_White, kde mělo nejvyšší číslo, a to celých 2352 kusů. Poté kleslo u stanice P\_Red na číslo 2004 kusů a u třetí stanice se lehce zvedlo na 2020 kusů.

Oproti tomu, jak Triangulární, Weibullovo tak Gamma rozdělení mělo první stanoviště nejnižší. Poté všechny 3 rozdělení měla nejvyšší počet kusů u stanice 2 P\_Red a naposledy střední hodnotu u třetí stanice P\_Blue.

Nejvíce kusů dle simulace ale bylo vygenerováno při exponenciálním rozdělení, kde největší rozdíl oproti ostatním je na prvním stanovišti P\_White.

Dalším krokem bylo dopočítání počtu kusů, které by reálný model vygeneroval, kdyby byl také puštěn po dobu dvě stě čtyřicet hodin. Zde by se vygenerovalo 2976 bílých puků, 2016 červených puků a 672 modrých puků.

Posledním krokem bylo porovnání kusů z distribučních funkcí a kusů z dopočtení. K tomu slouží tabulka 7-3. Pro porovnání, celkově při výpočtu by bylo vygenerováno 5664 kusů za čas dvě stě čtyřicet hodin. Žádná z distribučních funkcí se nedostala na stejné číslo. Nejbližší distribuční funkce, a to Triangel, k tomuto výsledku byla 5800 kusů, rozdíl tedy 136 kusů, o kterých vygenerovala více. Druhá pomyslná příčka by patřila dvěma distribučním funkcím, a to Weibull a Gama. Nejvíce kusů vygenerovala Negativně exponenciální distribuční funkce. Vypočtená hodnota, 5664 kusů, a hodnota Triangel distribuční rozdělení, 5800 kusů, se liší o 2,4%.

Tabulka 7-3: Porovnání výsledků experimentů

<b>Simulace [ks]</b>				
	P_White	P_Red	P_Blue	SUMA
Normal	negativni hodnoty			
Ngexp	2352	2004	2020	6376
Triangel	1739	2032	2029	5800
Weibull	1775	2059	1972	5806
Gamma	1802	2028	1976	5806
<b>Výpočet [ks]</b>				
	2976	2016	672	5664

## 8. ZÁVĚR

Cílem bylo propojit reálný model s modelem simulace a ověřit jeho věrnost. Dále na modelu simulace provést určité dlouhodobé testy, které by v reálném životě znamenaly desítky hodin čekání na data.

Jako prvním bodem proběhlo po teoretické stránce načerpání potřebných vědomostí, které vedly do praktické části.

V rámci diplomové práce byl vytvořen nový model výrobní linie ze stavebnice Fishertechnik, která se nachází na Západočeské Univerzitě, Fakultě strojní, Katedře průmyslového inženýrství a managementu. Celý model se musel vymodelovat v programu Tecnomatix Plant Simulation, který se používal i na samotnou simulaci. Při porovnání modelů nebylo dosaženo stoprocentní shody po grafické stránce z důvodu toho, že program nabízí pouze základní geometrické útvary. Naopak ale toto modelování pomohlo k pozdějšímu nastavování animací strojů, které se vytvářeli v prostředí programu. Po vymodelování přišlo na řadu programování a nastavení právě animací. Muselo se nastavit řízení aplikace v řídicí aplikaci pro propojení s modelem, kde se musela přepsat část názvů jednotlivých eventů pro propojení s modelem. Tímto se právě jednotlivé hodnoty (eventy) řídicí aplikace propojily se stroji a vytvořili digitální stín. Po spuštění aplikace se sesbírала data, která se dále použila pro testování nových distribučních funkcí na modelu. Také proběhlo ověření modelu a řídicí aplikace, jestli model simulace odpovídá realitě. Výsledek tohoto ověření byl pozitivní a mohlo se tedy přejít na experimenty dlouhodobé simulace. Model se tedy mohl přepnout na simulaci dlouhodobého chování. Pro toto nastavení proběhlo pět testů různých distribučních funkcí, kde každá funkce měla rozdílné výsledky. Bohužel u normálového rozdělení neproběhlo měření, protože mohlo dojít k záporným hodnotám. Zde program lehce selhal, že po analýze neposkytl dolní a horní mez. Nejpřesnější distribuční rozdělení bylo Triangel, které se nejvíce přiblížilo výsledkům, které se dopočítaly.

Samotný model simulace bude v průběhu času upravován a rozšiřován o nové části reálného modelu. Dále se na něm plánují dělat další simulace pro další experimenty..

## Seznam použité literatury

- [1] Dr. Simon Monk. *Programming Arduino: Getting Started with Sketches* (McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto, 2012).
- [2] I2C Bus Fundamentals. Texas Instruments [online]. 8 [cit. 2023-05-29]. Dostupné z: <https://www.ti.com/lit/an/slva704/slva704.pdf>
- [3] Dr. Simon Monk. *Programming the Raspberry Pi: Getting Started with Python, Second Edition* (McGraw-Hill Education: New York, Chicago, San Francisco, Athens, London, Madrid, Mexico City, Milan, New Delhi, Singapore, Sydney, Toronto, 2016).
- [4] GARDEN, Tracy, Elbrie DE KOCK a Tech AGE KID. *Micro: Bit in Wonderland: Coding & Craft with the BBC Micro:bit (microbit)*. Tech Age Kids, 2019. ISBN 1999787927, 9781999787929.
- [5] BANKS, A. *MQTT Essentials: A Lightweight IoT Protocol* [online]. 2014 [cit. 2023-05-29]. Dostupné z: <https://www.hivemq.com/mqtt-essentials>
- [6] Tecnomatix Plant Simulation [online]. Siemens Digital Industries Software, 2021 [cit. 2023-05-29]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/plant-simulation.html>
- [7] L. Wu et al., "Defect and Fault Modeling Framework for STT-MRAM Testing," in *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 2, pp. 707-723, 1 April-June 2021, doi: 10.1109/TETC.2019.2960375.
- [8] Tao, F., Cheng, J., Qi, Q. et al. Digital twin-driven product design, manufacturing and service with big data. *Int J Adv Manuf Technol* 94, 3563–3576 (2018). <https://doi.org/10.1007/s00170-017-0233-1>
- [9] Grieves, Michael & Vickers, John. (2017). Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. 10.1007/978-3-319-38756-7\_4.
- [10] Lu, Yuqian & Liu, Chao & Wang, Kevin & Huang, Huiyue & Xu, Xun. (2019). Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*. 61. 10.1016/j.rcim.2019.101837.
- [11] Qian, Cheng, Xing Liu, Colin Ripley, Mian Qian, Fan Liang, and Wei Yu. 2022. "Digital Twin—Cyber Replica of Physical Things: Architecture, Applications and Future Research Directions" *Future Internet* 14, no. 2: 64. <https://doi.org/10.3390/fi14020064>
- [12] Schrödel, M., & Steinhart, H. (2018). Design of digital laboratory experiments based on a Fisher Technik training system. *International Journal of Engineering Pedagogy*, 8(3), 111-120.
- [13] Hert, S., & Kucza, J. (2020). Impact of Using the VEX IQ Robotics System on Primary School Students' Attitudes Towards Technology and Engineering. *European Journal of Contemporary Education*, 19(4), 608-619
- [14] Basnet, S., & Crippen, K. J. (2018). Engaging students in science through robotics: Impact of robotics on science learning in elementary and middle schools. *Journal of Science Education and Technology*, 27(3), 269-283.

- [15] Schraeder, P., & Hölscher, M. (2019). New approaches to automation technology in mechatronics education. *International Journal of Engineering Pedagogy*, 9(1), 1-11.
- [16] Stage, Frances & Schulz, Phyllis. (2013). *STEM the Tide: Reforming Science, Technology, Engineering, and Math Education in America* by David E. Drew. *The Journal of Higher Education*. 84. 897-899. 10.1353/jhe.2013.0034.
- [17] Johnson, C.C., Mohr-Schroeder, M.J., Moore, T.J., & English, L.D. (Eds.). (2020). *Handbook of Research on STEM Education* (1st ed.). Routledge. <https://doi.org/10.4324/9780429021381>
- [18] Banks, J., Carson, J., Nelson, B. L., Nicol, D. (2004). *Discrete-Event System Simulation* (4th Edition). Prentice Hall. ISBN: 0131446797
- [19] Případ užití jako východisko pro digitální dvojče [online]. In: . [cit. 2023-05-29]. Dostupné z: <https://www.cagi.cz/upload/documents/konference/givs2021/21-GIVS-2021-Svoboda-Leos.pdf>
- [20] HEXBUG VEX Robotics – Stavební jeřáb [online]. In: . [cit. 2023-05-29]. Dostupné z: <https://www.promodels.cz/hexbug-vex-robotics-stavebni-jerab-p47589>
- [21] Lego Education [online]. In: . [cit. 2023-05-29]. Dostupné z: <https://stock.adobe.com/cz/images/minsk-belarus-may-2020-the-new-spike-prime-lego-robot-it-can-help-to-teach-children-programming-and-robotics-stem-and-steam-education-keep-education-yourself-concept/350555223>
- [22] Festo LT [online]. In: . [cit. 2023-05-29]. Dostupné z: [https://www.festo.com/lt/en/c/technical-training/learning-systems/industrial-trades-id\\_FDID\\_01\\_10/](https://www.festo.com/lt/en/c/technical-training/learning-systems/industrial-trades-id_FDID_01_10/)
- [23] Bangsow, Steffen. *Tecnomatix Plant Simulation: modeling and programming by means of examples*, Springer, 2020. ISBN 978-3-030-41544-0.
- [24] MALAGA, Miroslav a ULRYCH, Zdeněk. *Základy řízení robotů pro strojní inženýrství / autoři: Miroslav Malaga, Zdeněk Ulrych* [CD-ROM]. První vydání. Plzeň: Západočeská univerzita v Plzni, 2020. ISBN 978-80261-0486-5
- [25] Mařík, V. *Průmysl 4.0: Výzva pro Českou republiku*. Management Press, 2016. ISBN 978-807-2614-400
- [26] MANLIG, F. - KELLER, P.: Možnosti využití počítačové simulace. In: *Automatizácia technologickej prípravy výroby. Odborný seminár s medzinárodnou účasťou* Zvolen 17.09.1998, Zborník prednášok, s. 55..58
- [27] SEPPÄLÄ, Ludmila. Digital model, digital shadow, or digital twin – what is at the core of data-driven shipbuilding? *CADMATIC blog* [online]. 3 [cit. 2023-08-08]. Dostupné z: [https://www.cadmatic.com/en/resources/blog/digital-model-digital-shadow-or-digital-twin/?creative=517537585410&keyword=&matchtype=&network=g&device=c&gclid=Cj0KCQiA5aWOBhDMARIsAIXLlike1NFy0mTbnsrl13s82Rj0ae1FwQi6w-vrV6LAuB5e5YXKqIZUMF8gaAiIiEALw\\_wcB&fbclid=IwAR1qaKJkQhDB8utucUrlta02xDORyLHq8jxb1cCW-ue-mJ6fFhbFMiKPKkw](https://www.cadmatic.com/en/resources/blog/digital-model-digital-shadow-or-digital-twin/?creative=517537585410&keyword=&matchtype=&network=g&device=c&gclid=Cj0KCQiA5aWOBhDMARIsAIXLlike1NFy0mTbnsrl13s82Rj0ae1FwQi6w-vrV6LAuB5e5YXKqIZUMF8gaAiIiEALw_wcB&fbclid=IwAR1qaKJkQhDB8utucUrlta02xDORyLHq8jxb1cCW-ue-mJ6fFhbFMiKPKkw)
- [28] *Exploring the Concepts of Digital Twin, Digital Shadow, and Digital Model*. Oxford Insights [online]. 4 [cit. 2023-08-08]. Dostupné z:

[https://www.oxfordinsights.com/insights/2023/7/19/exploring-the-concepts-of-digital-twin-digital-shadow-and-digital-model?fbclid=IwAR1FzR50MAIgjv1xtxbPeOeWFZ6Ik0I3Ff\\_hPqDn4UvRrfGjP578-6YLhFQ](https://www.oxfordinsights.com/insights/2023/7/19/exploring-the-concepts-of-digital-twin-digital-shadow-and-digital-model?fbclid=IwAR1FzR50MAIgjv1xtxbPeOeWFZ6Ik0I3Ff_hPqDn4UvRrfGjP578-6YLhFQ)

[29] TLADI, Modisha. Digital Twins: What Are They and How Do They Work? [online]. 4 [cit. 2023-08-08]. Dostupné z: <https://www.makeuseof.com/digital-twins-what-are-they-and-how-do-they-work/>

[30] TAO, Fei; ZHANG, Meng; NEE, A. Y. C. Applications of digital twin. Digital Twin Driven Smart Manufacturing, 2019, 29-62.

## ELEKTRONICKÁ PŘÍLOHA č. 1

DP\_Veszpremi\_1.spp (simulační model)  
DP\_Veszpremi\_1.mset (nastavení řídicí aplikace)