



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## VIRTUÁLNÍ DVOJČE ROBOTICKÉ BUŇKY FANUC

VIRTUAL TWIN OF FANUC ROBOT CELL

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Robert Průdek

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2020



# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Robert Průdek

**ID:** 198026

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Virtuální dvojče robotické buňky FANUC

### POKYNY PRO VYPRACOVÁNÍ:

Vytvořte virtuální dvojče robotické buňky FANUC, navrhňte sadu laboratorních úloh.

- 1) Seznamte se s robotickou buňkou a dále vhodnými softwarovými nástroji pro realizaci virtuálního dvojčete buňky (jmenovitě Tecnomatix Process Simulate a Mechatronics Concept Designer), popište jednotlivé programy a zhodnoťte jejich vhodnost pro zamýšlené využití.
- 2) Ve zvoleném programovém prostředí realizujte kompletní fyzický model buňky, provedte propojení s virtuálním či fyzickým řídicím systémem.
- 3) Navrhňte laboratorní úlohu či úlohy o přiměřené obtížnosti, s pomocí kterých bude možnost rozšíření kapacity výuky. Realizujte návody k těmto úlohám a vzorové řešení
- 4) Vše otestujte a dokumentujte.

### DOPORUČENÁ LITERATURA:

Dokumentace k systému Tecnomatix Process Simulate

Dokumentace k systému Mechatronics Concept Designer

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

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

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

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

## **Abstrakt**

Tato bakalářská práce se zabývá vytvořením návrhu čelistí pro gripper firmy SCHUNK, čelisti byly vytvořeny pro čtyři dodané modely. Vytvoření simulace s těmito modely, kde dojde k přesunu daného modelu z jedné strany na druhou ve virtuálním dvojčeti buňky FANUC. Je zde popsán postup práce, který byl dodržen pro vytvoření programu pro robota.

## **Klíčová slova**

Tecnomatix Process Simulate, TPS, ROBOGUIDE, průmysl 4.0, NX, Mechatronic Concept Designer, MCD

## **Abstract**

This bachelor's thesis deals with design jaws for gripper from SCHUNK, jaws were made for four given models. Created simulation shifts model from one side to another in virtual twin of FANUC cell. Procedure, which was followed to create program for robot is described here.

## **Keywords**

Tecnomatix Process Simulate, TPS, ROBOGUIDE, industry 4.0, NX, Mechatronic Concept Designer, MCD

## **Bibliografická citace:**

PRŮDEK, Robert. *Virtuální dvojče robotické buňky FANUC* [online]. Brno, 2020 [cit. 2020-06-06]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/126921>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Václav Kaczmarczyk.



## **Prohlášení**

Prohlašuji, že svou bakalářskou práci na téma „Virtuální dvojče robotické buňky FANUC“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

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

V Brně dne: **8. června 2020**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing Václavu Kaczmarczykovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Dále chci poděkovat panu Ěng. Vítu Přichystalovi, který mi poskytl technickou podporu z firmy AXIOM TECH s.r.o..

V Brně dne: **8. června 2020**

.....  
podpis autora

# Obsah

1	Cíl práce .....	9
2	Průmysl 4.0 .....	10
2.1	Revoluce v průmyslu .....	10
2.2	Důležité pojmy .....	10
2.3	Základní principy .....	11
2.4	Chytrá továrna .....	13
3	Robotická buňka od firmy FANUC .....	14
3.1	ROBOGUIDE .....	14
4	Práce v NX.....	15
4.1	NX.....	15
	Příruba.....	16
4.2	Návrh čelistí .....	17
4.3	Sjednocení .....	18
4.4	Dodatečné práce na modelu buňky .....	18
5	Porovnání softwarů .....	20
5.1	Mechatronic Concept Designer .....	20
5.2	Tecnomatix Process Simulate .....	20
5.3	Volba programu.....	22
6	Tvorba laboratorních úloh.....	23
6.1	Tvorba hliníkového profilu .....	23
6.2	Seskládání buňky z hliníkových profilů .....	23
6.3	Tvorba programu na reálném robotovi .....	24
6.4	Úpravy starého zadání.....	25
6.5	Robot v buňce a jeho simulace .....	26
7	Závěr.....	35
	Seznam symbolů, veličin a zkratk.....	39
	Seznam příloh.....	40

## Seznam obrázků

Obr. 1 Obrázek průmyslových revolucí [1] .....	10
Obr. 2 Model RAMI 4.0 [3] .....	11
Obr. 3 Pracovní plocha programu NX.....	16
Obr. 4 Příruka pro spojení gripperu s robotem.....	16
Obr. 5 Vytvořené čelisti.....	17
Obr. 6 Dodané modely .....	17
Obr. 7 Kompletní gripper .....	18
Obr. 8 Buňka s obrysem robota .....	19
Obr. 9 Mechatronic Concept Designer.....	20
Obr. 10 Tecnomatix Process Simulate.....	21
Obr. 11 Hliníkový profil .....	23
Obr. 12 Poskládaný rám z hliníkových profilů .....	23
Obr. 13 Teach pendant.....	24
Obr. 14 Gripper rozdělen na pohyblivé a nepohyblivé části .....	25
Obr. 15 Gripper po definici TCPF .....	26
Obr. 16 Posunutý model na první místo .....	27
Obr. 17 Souřadnicový systém pro Pick and Place operaci .....	28
Obr. 18 Nastavení operace Pick and Place .....	29
Obr. 19 Usazení robota funkcí Smart Place .....	30
Obr. 20 Kompletní virtuální simulace přesunu modelu.....	31
Obr. 21 Chyby viditelné v Robot Program Viewer .....	32
Obr. 22 Definice Uframe a Utool.....	33
Obr. 23 Definování signálů pro robota.....	33
Obr. 24 Buňka v programu ROBOGUIDE.....	34

# 1 CÍL PRÁCE

Jedním z hlavních cílů této práce bylo vytvoření nebo rozšíření jedné či více laboratorních úloh do laboratorních cvičení pro studenty.

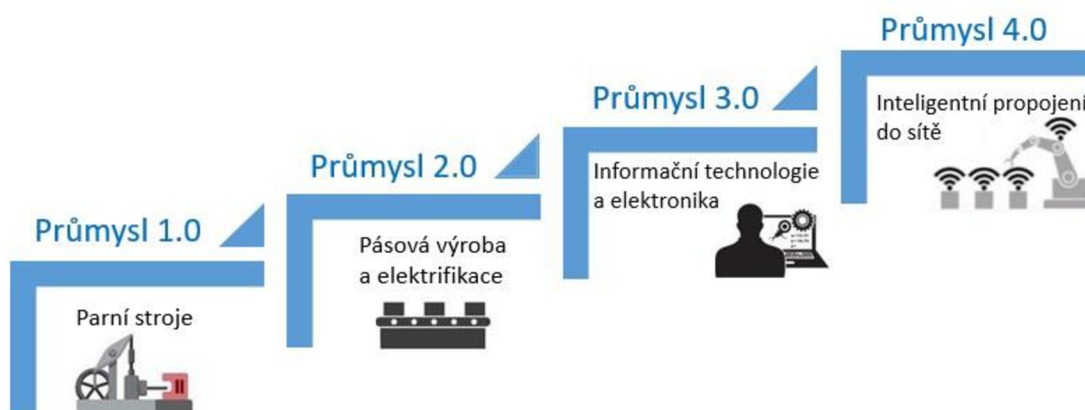
Dále vytvoření funkční simulace robota FANUC, který přemístí daný objekt z jedné strany pracovní desky na druhou. Pro možnost sestavení této simulace se nabízely dva programy. Prvním programem je Tecnomatix Process Simulate a druhý Mechatronic Concept Designer. Následovalo seznámení se s oběma programy a poté rozhodnutí, který by pro danou aplikaci byl vhodnější. Zvolený program měl mít možnost vyexportovat vytvořené pohyby robota, aby je bylo následně možné nahrát do fyzického robota a otestovat správnost vytvořené simulace.

Poslední částí práce bylo vytvořit dokumentaci pro zvolené řešení a zaimplementovat ho do návodu na cvičení Průmyslu 4.0. Zadokumentované řešení bude využíváno jako laboratorní úloha pro studenty, kteří díky tomu dostanou možnost vyzkoušet si virtuální zprovoznění robota a následně tento jejich program do něj nahrát, tím zjistí, jestli jejich návrh je vhodný či ne.

## 2 PRŮMYSL 4.0

### 2.1 Revoluce v průmyslu

Za první revoluci v průmyslu se dá považovat vynález parního stroje. Zde je možné zahrnout tkací stroj poháněný párou, který vynalezl Edmund Cartwright [1]. Druhá revoluce nastala za pomoci elektrické energie, začaly se rozmáhat elektrické montážní linky. Urychlení průmyslové revoluce pomohl Thomas Edison a jeho vynález žárovky. Třetí revoluci odstartovalo vyrobení prvního programovatelného logického automatu. Myšlenka čtvrté revoluce je kompletním propojením všech zařízení, osob, systému a subsystémů v továrně, vznikne tím takzvaná „chytrá továrna“. [2]



Obr. 1 Obrázek průmyslových revolucí [1]

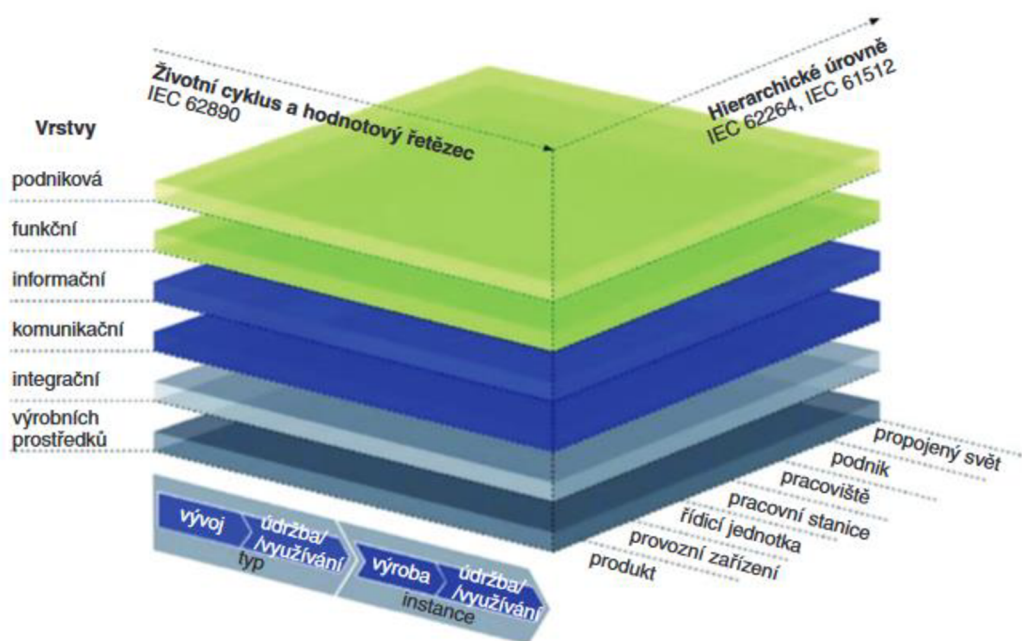
### 2.2 Důležité pojmy

**Internet věcí (IoT)** – síť, ve které jsou připojena všechna zařízení v továrně, jako jsou například roboti, dopravníky, 3D tiskárny a další. Může to být i chytrá domácnost, kde jsou zapojeny různé domácí spotřebiče. Komunikace je možná nejen s člověkem, ale i mezi stroji. V továrně jsou fyzické objekty vybaveny senzory s vestavěným mikropočítačem a připojením k internetu. Díky tomu jsou data předzpracována a odeslána, je zde i možnost přijímání dat.

**Digitální dvojče** – fyzický model převedený do digitální formy. Slouží jako propojení fyzického světa s tím digitálním. Zrcadlí život daného fyzického dvojčete. Skládá se ze tří částí, první je fyzický produkt, další je digitální část a poslední je spojení mezi fyzickým a digitálním produktem.

**Kyberneticko-fyzikální systém (CPS)** – spojuje fyzikální systémy s digitálními modely pomocí IoT. Je vybaven jednoduchou inteligencí, díky tomu se může rozhodovat na základě jisté logiky a může komunikovat s dalšími CPS.

**RAMI 4.0** – Autory RAMI modelu jsou BITCOM, VDI/VDE a ZVEI. Tento model slouží pro popis celé struktury průmyslu 4.0 v trojrozměrném prostoru. Tato předloha reprezentuje, co si pod pojmem průmysl 4.0 představuje Německo. Každé průmyslové zařízení, je možné díky tomuto modelu snadno klasifikovat kam patří. Při návrhu zařízení je možné stanovit, na které vrstvy na všech osách bude zařízení implementováno. Lze stanovit potřebnou míru interoperability nového zařízení, zajistit jednoduchou syntaxi a sémantiku. Je možné definovat kritické služby a jejich požadovanou kvalitu, přesnost synchronizace dat, schopnost práce v reálném čase a další.



Obr. 2 Model RAMI 4.0 [3]

**Asset administration shell (AAS)** – připojuje konkrétní zařízení do průmyslu 4.0, aby s ním mohlo být komunikováno, je to tedy standardizované komunikační rozhraní. Ukládá všechna data a informace o stroji, pro který je vytvořena tato schránka. Každé fyzické zařízení má svůj vlastní AAS. Několik zařízení může tvořit jeden funkční celek se společným AAS. [2]

## 2.3 Základní principy

**Interoperabilita** – je schopnost všech systémů, například CPS a lidí, vyměňovat a sdílet informace s pomocí syntaktických nebo sémantických pravidel. Syntaxe se zabývá tím, aby byla definovaná syntaxe a kódování, jak pro vyšší programové jazyky, tak pro ty nižší. Nezabývá se kontextem. Kontext spadá pod sémantickou interoperabilitu, která se snaží smysluplně a přesně interpretovat význam přenášených informací napříč různými systémy. Interoperabilita má dvě

podoby, vertikální a horizontální. Vertikální vede od signálů, plánování, rozhodování výroby a plánování podnikových zdrojů, neboli ERP až po rozhodování na nejvyšší úrovni. Patří sem komunikace stroj se strojem, v tomto bodě vzniká jeden z největších problémů, protože každý výrobce používá jiné komunikační standardy. Bylo by tedy nutné implementovat vrstvu pro porozumění syntaxi a sémantiku. Jedno z řešení by bylo, aby všichni výrobci dodrželi jednotný standard nebo využití otevřených standardů, jako jsou například OPC-DA, OPC-HDA, OPC-A&E, OPC-UA. Horizontální integrace je kompletní počítačové propojení na všech úrovních. Tím je myšleno podání objednávky zákazníka, zásobovací řetězec, vývoj, výrobu, expedici a distribuci.

**Virtualizace** – je schopnost nahrazení fyzických prototypů virtuálními a simulačními nástroji. Vytvoří se virtuální výrobky, výrobní prostředky a výrobní procesy, s těmi se pak dále pracuje. Jde sem zařadit i virtuální realitu. To je počítačově vytvořený umělý prostor nebo jeho reálná kopie. Využití najde při plánování umístění výrobních linek a ostatních pracovišť. Tato realita se nejčastěji spojuje s virtuálními brýlemi, kde se člověk cítí, jako by tam právě byl. To je možné využít pro simulace. Dále je tady rozšířená realita, ta může například pomocí tabletu ukázat něco, co je potřeba udělat nebo poskytně informace o objektu, který je středem zájmu. Například dojde k požadavku povolit ventil pro uvolnění páry. Pokud se nachází pracovník v dané místnosti poprvé, je složité zjistit, která z mnoha páček by měla být ta správná. S pomocí tabletu namíří na všechny páčky a díky tomu získá dostatek informací potřebných pro zvolení správné páčky. Poslední je Light Guide System, při montáži daného produktu tento systém vede pracovníka, jak dále postupovat, které součástky má přiložit pro kompletaci, na která místa vložit a zašroubovat šroubky a tak dále. Kdyby došlo k implementaci tohoto systému do helmy se zobrazovacím sklem, bylo by možné do jisté míry nahradit nutnost rozšířené reality. Pro představu nastal stejný problém jako výše s páčkou pro uvolnění páry. Operátor vyšle požadavek, že potřebuje někoho, aby došel do místnosti s touto páčkou a povolil ji. Člověk, který to má na starost dostane požadavek, kdyby náhodou neznal cestu do místnosti, helma se zobrazovacím sklem ho navede na správné místo a ukáže páčku, kterou je třeba povolit.

**Decentralizace** – rozhodování nespadá pod jeden velký uzel, nýbrž každá část subsystému si sama rozhoduje a řídí, co bude následovat dále. Data jsou decentralizována také, komunikace je řešena pomocí IoT.

**Schopnost pracovat v reálném čase** – dostatečně rychlé rozhodování a řízení v reálném čase je podmínkou pro libovolnou komunikaci. Reálný čas ve výrobě znamená, že doba komunikační odezvy druhé strany je rychlejší než výrobní doba. Ve virtuálním modelu to znamená, že reakce technologie jsou dostatečně rychle přenášeny do modelu.



**Orientace na služby (SOA)** – kdy jsou jednoznačně preferovány filozofie nabízení a využívání standardních služeb. Poskládání různých aplikací pomocí komponent poskytujících služby, které na sobě nejsou závislé.

**Modularita a rekonfigurabilita** – je výhodné, aby systémy v průmyslu 4.0 byly modulární a byly schopny autonomní rekonfigurace. Například na základě nově pokaženého stroje vyhodnotit situaci a začít ji řešit. [2]

## 2.4 Chytrá továrna

Typ výroby v chytré továrně je nejlepší si představit na automobilové výrobě. V dnešní době se auta skládají se vším všudy. Například elektrické kabely, které se dávají do auta jsou stejné u auta v plné, nejdražší výbavě i v nejlevnější, očesané. Rozdíl mezi auty je ten, že pro levnější verzi se některé kabely nevyužijí, ale jsou namontovány. V chytré továrně to bude probíhat trochu jinak. Zákazník dá požadavek na auto, které chce a v jaké výbavě. Továrna zaeviduje tuto zakázku a vytiskne například RFID čip, na kterém budou informace důležité pro výrobu auta zákazníkovi. Tento čip dostane například vozík, zeptá se, jestli je někde ta kovová konstrukce, kterou potřebuje. Dostane odpověď, že ve skladu tuto konstrukci mají. Dotáže se, jestli je někdo, kdo by mu konstrukci dovezl. Ozve se vozík, který má čas, a doveze danou komponentu. Jelikož vozík ví, že následuje natažení kabelů, zeptá se kdo to umí. Dostane odpověď od různých stanic, které to umí a mají čas. Na základě vzdálenosti si vybere, co je pro něj nejvýhodnější. Dojede ke stanici a sdělí jaké kabely a kam se mají natáhnout. Jelikož vozík ví, že po kabelech bude potřebovat karoserii vytisknutou na 3D tiskárně, zeptá se a následně zažádá nejbližší 3D tiskárnu ke stanici natahující kabely, aby nedocházelo k dlouhému čekání, než se vytisknou potřebné díly. Dále pro specifické úkony by se dal využít Light Guide systém, který již byl zmíněn.

## **3 ROBOTICKÁ BUŇKA OD FIRMY FANUC**

Firma FANUC nabízí pro školy možné balíčky, různý software a jeho kombinace až po hlavní balíček, kde je vše včetně buňky s robotem a potřebnými komponenty pro jeho provoz. Jedním z těchto programů je ROBOGUIDE. [4]

Tahle buňka najde velké užití pro studenty, kteří se díky tomu seznámí se základy ovládání robotů. Je možné si stáhnout ze stránek firmy velice podrobný manuál, který naučí nezkušeného uživatele všem potřebným základům k ovládání buňky. Robot je ošetřen i proti neúmyslné chybě uživatele, kdy v momentě nárazu, tedy vzniku velkého odporu vůči některému z kloubů, oznámí chybu a zastaví se. Pro další možnost posunu je nutné chybu vyresetovat a je možné pokračovat dále.

### **3.1 ROBOGUIDE**

Jedná se o simulátor robota, který dokáže nasimulovat pohyby robota i aplikační příkazy, tím se výrazně zkrátí doba vytváření nových konfigurací pohybu. Návrh, testování i úpravy je možno provádět ve vypnutém režimu. Program má k dispozici velkou knihovnu všech FANUC robotů, strojů a jednoduchých nástrojů. Je možné vybrat a vyměnit jeden díl za druhý, popřípadě změnit jeho velikost. Program je navržen tak, aby byl jednoduchý a zaškolení na jeho používání vyžadovalo co nejméně času. Programování v něm je stejné jako programování pomocí teach pendantu. [5] Tento program je možné využít pro převod programu ve formátu ls nebo pe do takového, se kterým daný robot dokáže pracovat, tím formátem je tp.

## 4 PRÁCE V NX

### 4.1 NX

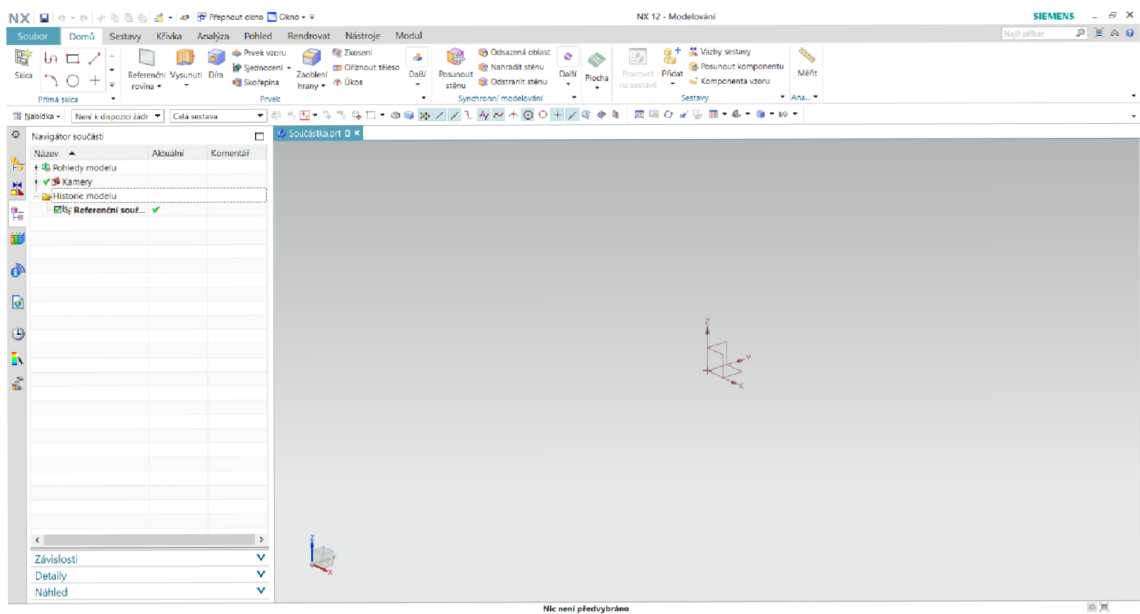
Program NX patří do skupiny Siemens PLM Software od firmy Siemens. Do Siemens PLM patří i například Tecnomatix, Teamcenter, Simcenter a další. [6] NX je 3D CAD modelovací systém, kde za pomoci 2D skic je možné vytvořit různě složité 3D objekty. [7] Těm je možné následně přiřadit barvy a různé materiály, jako jsou kovy, sklo a další. Zvládá vytvářet simulace, výpočty, analýzy, obrábění a jeho následnou simulaci. [8]

Program má plnou asociativitu všech spolupracujících modulů, postavenou nad jednotnou, grafickou, objektově orientovanou databází. Díky tomu je možné na modelu provádět pevnostní a kinematické výpočty, popřípadě simulace a analýzy. Je zde využit takzvaný Master Model Koncept, který zajišťuje, že změny provedené na výrobku se přenesou do všech rozpracovaných aplikací. Přímo v NX je tohoto využíváno. Pokud je v NX několik dílů, které jsou vloženy k dalším, nebo jsou vytvořeny jako sestava z několika dílů, tak změny provedené mimo sestavu přímo na dané součástce se projeví změnou v sestavě. [9]

Okno programu, po vytvoření nového souboru typu model, se dá rozdělit na 3 hlavní části. První je hlavní nabídka, kde se nachází záložky domů, sestava, křivka, analýza a tak dále. Přibližně 70 % práce lze zvládnout ze záložky domů. Možnost modifikace domovské záložky je velice užitečná. Uživatel si zde přidá funkce, které jsou používány nejčastěji a odebere nepoužívané, tím zařídí to, že vše má na jednom místě a při práci se zbytečně nezdržuje.

Další částí je levý panel, kde se nachází 3 hlavní sekce. Nejpoužívanější bude rozhodně Navigátor součásti. Tady lze najít vše od vytvořených skic, vysunutí po zkosení a tak dále. Další je Navigátor sestavy. Zde je daná součástka, na které se právě pracuje, a pokud dojde ke vložení jiné již vytvořené součástky, objeví se v tomto seznamu. Poslední sekce je Navigátor vazby. Ten zobrazuje všechny vytvořené vazby mezi vloženými komponenty. A poslední z hlavních částí je kreslená plocha, kde se vytváří všechny 3D modely. Program se mnoha funkcemi

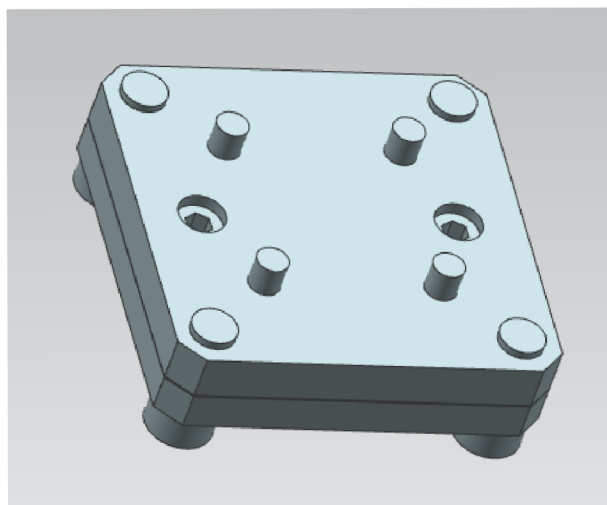
podobá jiným modelovacím programům, avšak někdy jsou funkce nazvané jinak než by daná osoba čekala.



Obr. 3 Pracovní plocha programu NX

## Příruba

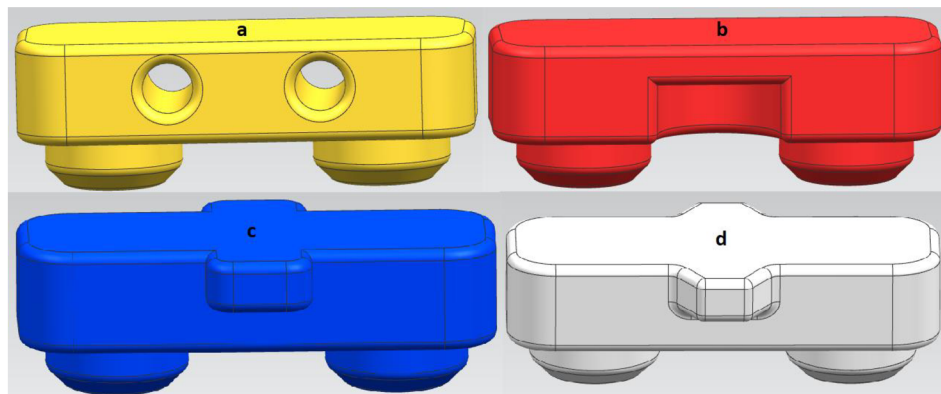
Jelikož gripper je nutné nějakým způsobem spojit s robotem, nemohla ani tato součástka chybět. Díky tomu, že fyzický robot je k dispozici bylo možné změřit potřebné míry pro možnost vymodelovat tuto komponentu a následně ji spojit s celým gripperem.



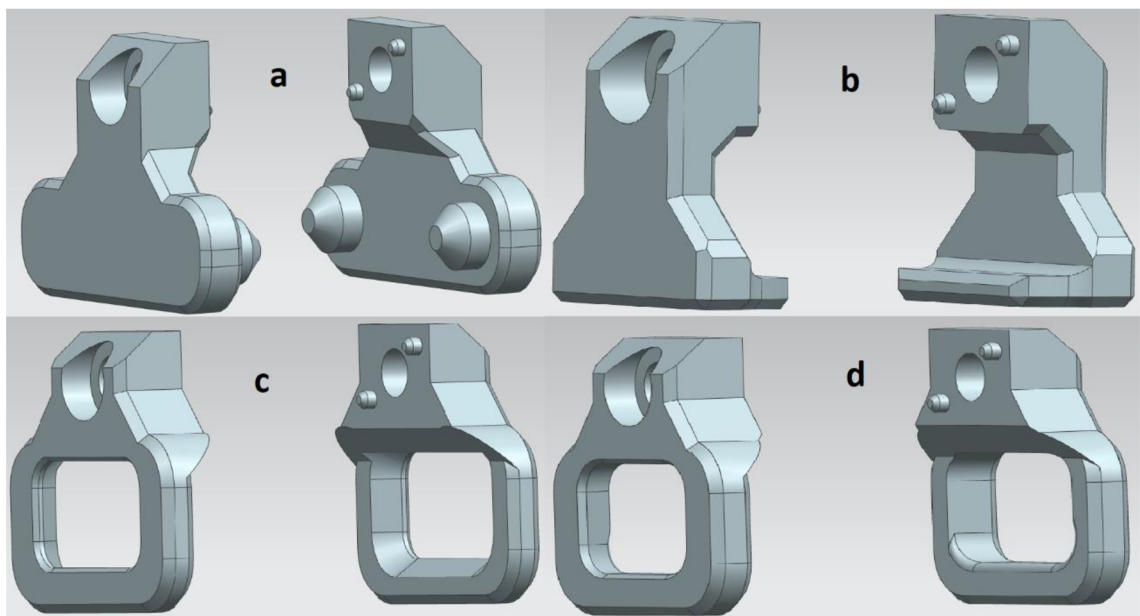
Obr. 4 Příruba pro spojení gripperu s robotem

## 4.2 Návrh čelistí

Vedoucí této práce dodal 4 modely, pro které měly být zhotoveny čelisti, pro něž by se následně vytvořila laboratorní úloha pro další studenty. Jako první bylo nutné získat model gripperu, který je upevněn na fyzickém robotovi. V tomhle případě se jedná o uchopovací systém firmy Schunk, daný model je EGP 40-N-N-B. Virtuální model gripperu je možné získat na stránkách daného výrobce. Návrhy vytvořené v prostředí NX byly odsouhlaseny vedoucím práce, nebo upraveny tak, aby byly dostatečně vyhovující.



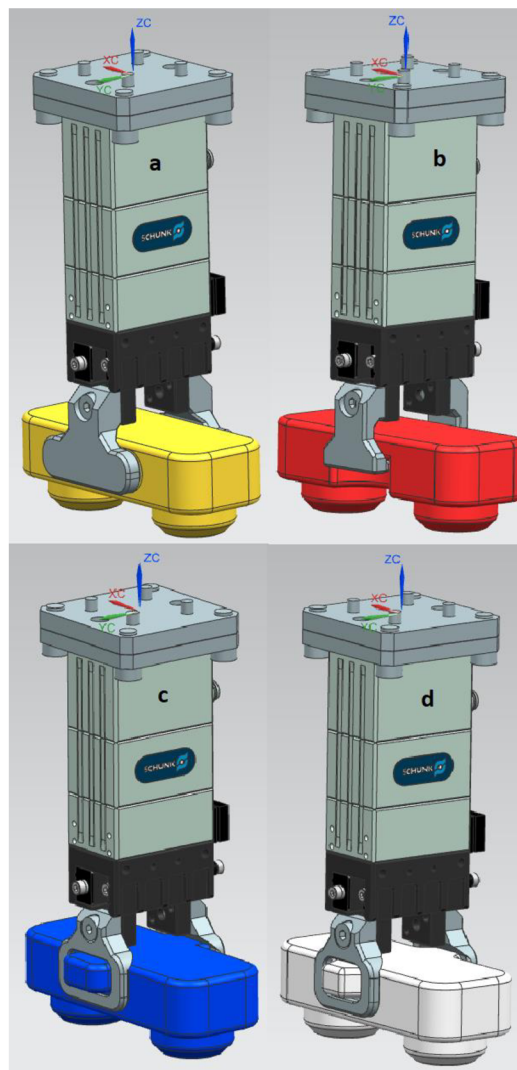
Obr. 6 Dodané modely



Obr. 5 Vytvořené čelisti

## 4.3 Sjednocení

Poslední částí je spojit vše do jednoho celku, následně vyexportovat do formátu jt pro možnost další práce v programu TPS. Důležitou věcí je to, aby počátek souřadnicového systému byl ve středu a ležel na té části, která se bude dotýkat robota, v tomto případě se jedná o přírubu. Kdyby tomu tak nebylo, nastal by problém při pokusu připojit gripper k robotovi. Směr os tohoto souřadnicového systému není v tomto případě důležitý, protože je možná korekce této chyby při spojení robota s gripperem.

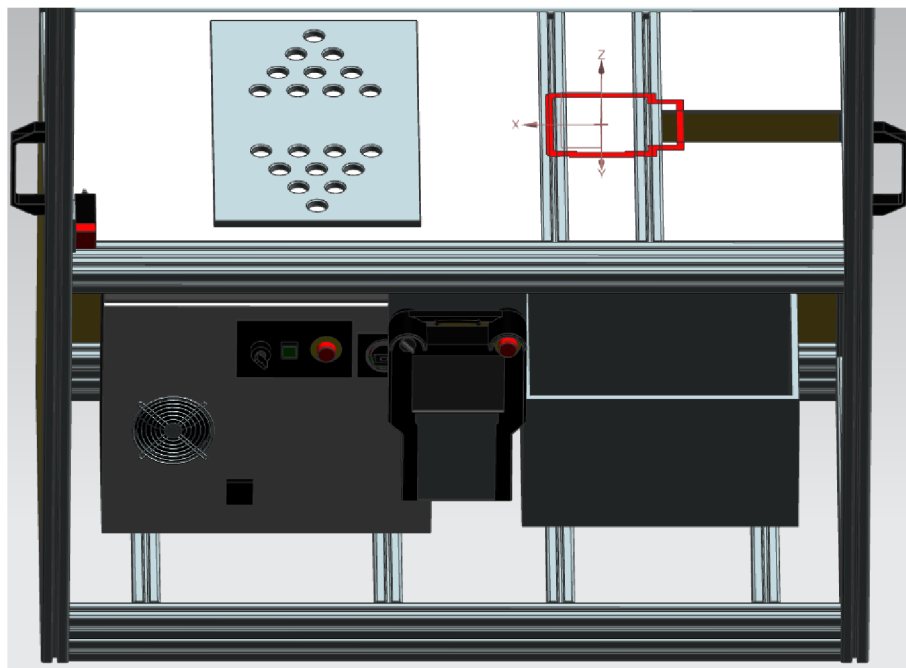


Obr. 7 Kompletní gripper

## 4.4 Dodatečné práce na modelu buňky

V semestrální práci došlo k převedení fyzické buňky do virtuální podoby. Až při práci s danou buňkou v programu TPS byly zjištěny problémy, u kterých došlo k odstranění. Problémem nastal v tom, že po vložení robota nebylo jisté, zda je na

správném místě či nikoliv. Tento problém byl vyřešen vytvořením obrysu podstavy robota, který nyní ukazuje jeho správnou pozici. Pro dokonalou přesnost je uprostřed tohoto objektu souřadnicový systém, pomocí něhož je snadné posunout robota na přesnou pozici. Jelikož na tento objekt se postaví daný robot, mohlo by dojít ke zvětšení vzdálenosti robota od zvedané součástky. Následným problémem by mohl být to, že fyzický robot se pokusí zvednout součástku, která je ve skutečnosti v jiné vzdálenosti, než bylo vypočteno v programu. Z tohoto důvodu je tloušťka tohoto objektu nastavena na 0,01 mm, pro účely potřebné k výuce je tahle hodnota zanedbatelná.



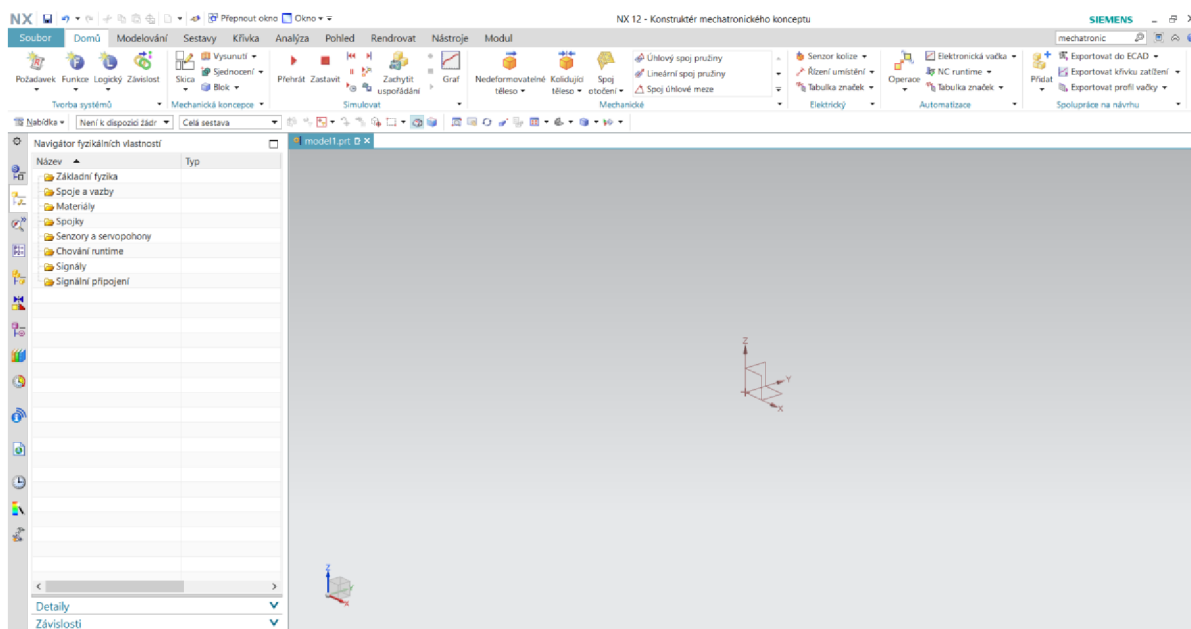
**Obr. 8 Buňka s obrysem robota**

Při následném nahrání již vygenerovaného programu a spuštění na fyzickém robotovi bylo zjištěno, že robot, když najede na pozici, kde má být model položen je zde odchylka jeden centimetr mimo požadovanou hodnotu. Tato odchylka byla způsobena špatným návrhem děrované desky pro modely. Při návrh této desky byly hodnoty zjištěny z exportované součást z programu ROBOGUIDE. Při zjištění chyby došlo kzměření správných hodnot, ze který plyne, že exportovaná součástka z programu neodpovídá skutečnosti. Následná náprava této chyby je předělat tuto součástku tak, aby odpovídala té, která je ve fyzické buňce.

# 5 POROVNÁNÍ SOFTWARE

## 5.1 Mechatronic Concept Designer

První program, který by bylo možné využít pro realizaci dané úlohy je Mechatronic Concept Designer, dále již jen MCD. Jedná se o modul programu. Další výhodou těchto modulů je jistá možnost propojení spolupráce konstruktérů, kteří se zabývají 3D designem a kinematikou, elektroinženýrů, kteří mají na starost pozice senzorů a pohonů, a programátorů, jež se postarají o chod daného modelu na základě nějakého řídicího kódu [10]. Díky propojenosti je možné řešit dané problémy rychleji. Není nutné svolávat každý den zasedání, kde by docházelo k probírání daných chyb. Program má možnost komunikovat online s externími HW zařízeními za pomoci protokolu OPC, popřípadě exportovat posloupnost operací ve formátu PLCopen. Pokud je použit OPC, potom se MCD chová jako klient. [11]



Obr. 9 Mechatronic Concept Designer

## 5.2 Tecnomatix Process Simulate

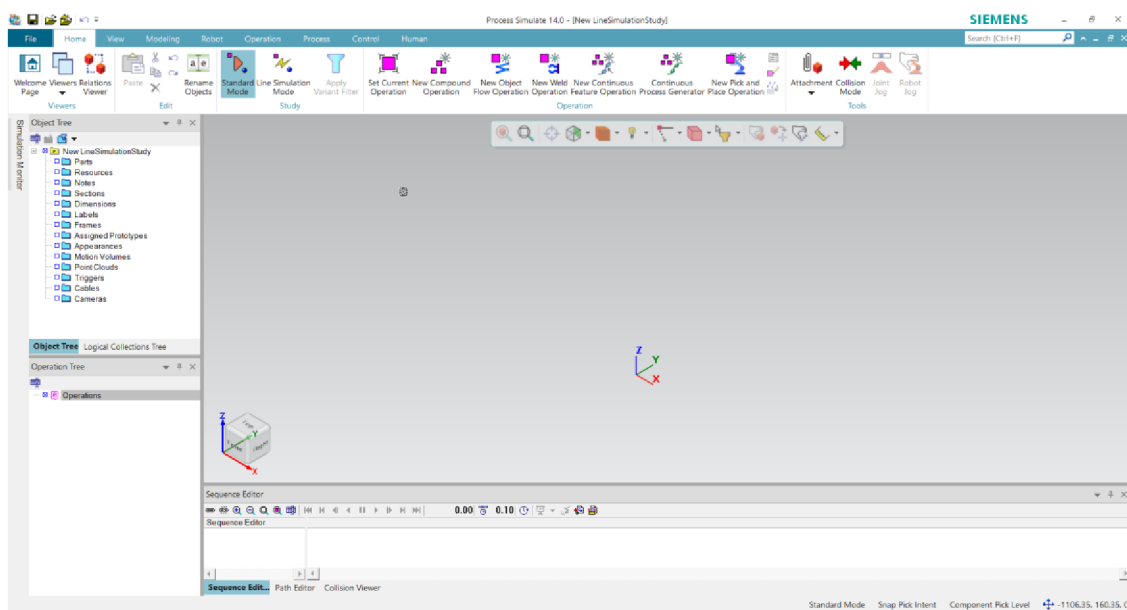
Tento program se dělí na více částí. Jedna pracuje s modely lidí a další s modely robotů a jiných strojů. Jeden z hlavních balíčků je Process Simulate Human. Používá se k digitálnímu modelování člověkem prováděných úkonů a jejich realistickou simulaci. Díky tomu je možné ověřit ergonomické standardy, bezpečnost práce a výkonnost.

Je zde mnoho analytických nástrojů pro zatížení člověka, z nich je možné se dozvědět důležité věci. Například pomocí analýzy pracovního prostředí, zda lidský



pracovník vidí vše, co vidět potřebuje a má. Že dosáhne na všechny nutná místa a unese všechny zvedané věci. Produkt cílí především na výrobní podniky a konzultační společnosti, které potřebují řešit výše zmíněné problematiky. Práce ve výrobě může být někdy náročná z pohledu pracovních poloh. Analýzy odhalí přetížení některých částí těla, únavy nebo rizika zranění. Nespornou výhodou je i normování časů. To se ve firmě dá použít jako přibližný odhad pracovní normy. [12]

Další hlavní částí je Process Simulate Robotics. Jedná se o digitální simulační nástroj pro 3D zprovoznění robotů ve výrobním procesu. Toto prostředí je možné propojit s dalšími programy od této firmy, jako jsou například Process Designer a Teamcenter Manufacturing. Lze zde navrhnout virtuální model celé výrobní linky, simulovat signály řídicích jednotek, kontrolovat proveditelnost montáže, kontrolovat ergonomii a ověřit funkčnost návrhu a zapojení. V programu je velká škála robotů, kteří zvládají všechny možné úkony od Pick&Place, balení, montáže, přes bodové, obloukové a laserové sváření, až po frézování, odjehlování, lepení, broušení, leštění, lakování a řezání laserem i vodním paprskem. Zde vytvořené programy lze nahrát do robotů, ale i stáhnout a načíst do PS, kde může být následně upravován. [13]



**Obr. 10 Tecnomatix Process Simulate**

## **5.3 Volba programu**

Po jednoduchém seznámení s programem MCD bylo zjištěno, že programu je nutné udat, jaké části se mohou pohybovat a směr jejich pohybu. Dodaný robot má šest kloubů, za jejichž pomoci se robot pohybuje v prostoru. Jelikož má být výstupem vytvořit úlohu pro studenty, kterou by mohli následně zopakovat, zdál se tento program jako velice nevyhovující. Po krátké konzultaci s pracovníky firmy AXIOM TECH, byl důrazně doporučen program TPS, tento program je velice užitečný pro práci s mnoha roboty.

## 6 TVORBA LABORATORNÍCH ÚLOH

### 6.1 Tvorba hliníkového profilu

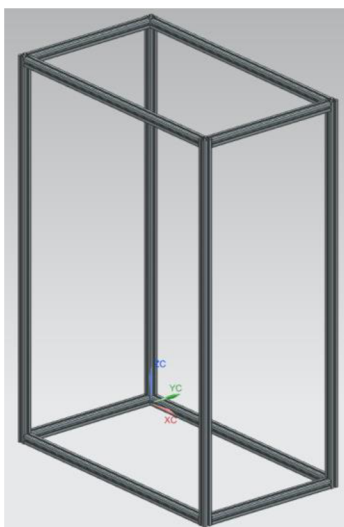
První úlohou, kterou si vyzkouší studenti je vytvoření hliníkového profilu 40x40 mm. Je to jednoduchá úloha, ve které se studenti seznámí s programem NX. Naučí se používat skicu a nástroje úzce s ní spojené.



Obr. 11 Hliníkový profil

### 6.2 Seskládání buňky z hliníkových profilů

Další částí cvičení je vytvořit tři velikosti profilů, z nichž se dá následně sestavit zjednodušený model buňky, jejíž fyzický model je v laboratoři. V tomto cvičení je snaha studenty naučit používat všechny různé vazby. Naučit je jistým rozdílem mezi nimi.



Obr. 12 Poskládaný rám z hliníkových profilů

## 6.3 Tvorba programu na reálném robotovi

Když studenti vytvoří model čelistí, je na čase vyzkoušet, jestli návrh jejich čelistí je správný a funkční. Naučí se zde manipulovat s robotem, nastavit Utool, to je kartézský koordinační systém, který definuje střed a směr nástroje. Dále zde budou nuceni definovat Uframe, ten definuje souřadnicový systém, který je posunutý od toho, který je pro robota výchozí. Výchozí je pro robota takzvaný světový souřadnicový systém, ten se nachází uvnitř druhého kloubu. Všechny body programu jsou potom k danému Uframe svázány, protože robot hodnoty koordinátu programu přepočítá na základě toho, který byl zvolen. Když se změní situace a deska, se kterou bylo pracováno změní svoje natočení vůči robotovi o několik stupňů, program přestane fungovat. Avšak pokud byl program svázán s nějakým Uframe, stačí daný Uframe znovu definovat a program začne bez problémů fungovat.

Dále si osvojí další práci spojenou s ovládací komponentou, která je tech pendant. Naučí se ovládat pohyby robota, zjistí, jak jsou pohyby odlišné na základě volby pohybu. Pohyb robota je možné volit například joint, kdy dochází k pohybu od j1, což je kloub pro otáčení celým robotem až po kloub j6, který se používá pro otáčení gripperu kolem středu jointu j6. Další možností je pohyb v souřadnicích nástroje, to je velice užitečné při přesném najetí k modelu. Další jsou například pohyby vůči světu, tím je myšleno vzhledem k jeho světově definovaným souřadnicím, které se nacházejí v druhém kloubu.

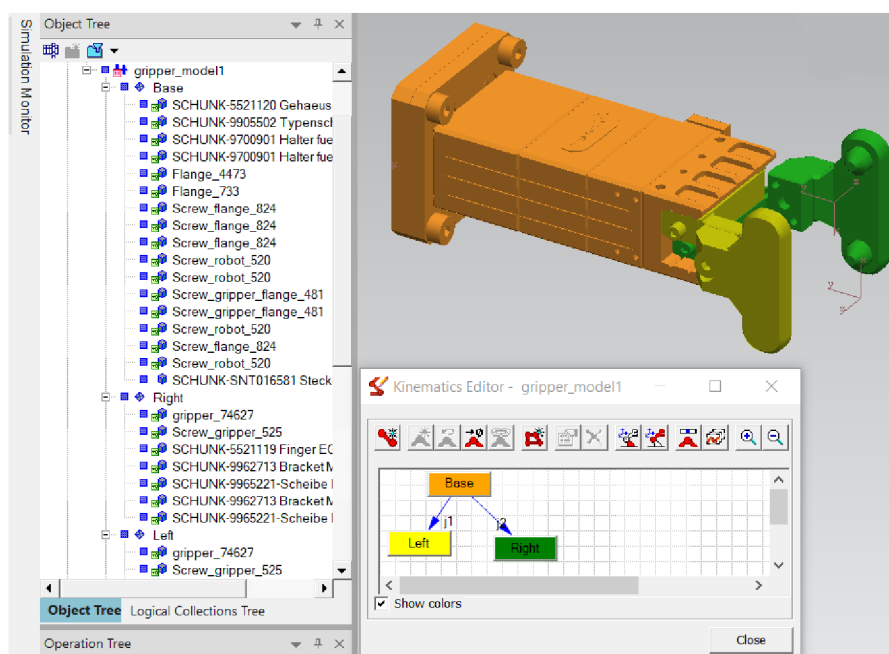


Obr. 13 Teach pendant

## 6.4 Úpravy starého zadání

Jelikož laboratorní úloha byla již jistým směrem rozpracována, bylo nutné ji mírně předělat, aby bylo možné pokračovat s nově vytvořenými komponentami. Jedna z těchto zmiňovaných komponent je příruba. S ní předešlá laboratorní úloha nepočítala, proto na tomto místě došlo k prvnímu výraznému zásahu do již zavedené laboratorní úlohy.

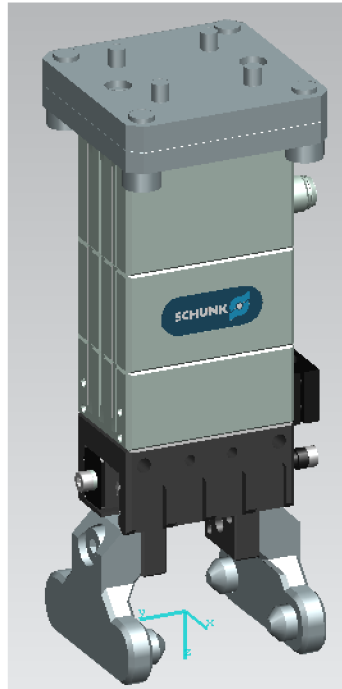
Šrouby pro upevnění čelistí ke gripperu jsou taky novinkou, kterou je nutno zakomponovat. Kdyby tomu tak nebylo, došlo by k tomu, že při posunu čelistmi do stavu otevřeno šroub zůstane na místě, ale čelisti a všechny jeho pohyblivé části se pohnou. Tahle věc je spíše kosmetická, ale pokud chce dotyčná osoba ukázat zákazníkovi nějaký produkt, o který si zažádal, je snaha o eliminaci všech možných chyb.



Obr. 14 Gripper rozdělen na pohyblivé a nepohyblivé části

Největší změnou si prošla část zaměřená na definici tool center point frame vytvořených čelistí, dále již jen TCPF. Zde je záměr naučit studenty více s programem NX, ve kterém namodelovali čelisti k přidělenému modelu. V NX se nachází funkce Měřit, díky ní je možné změřit vzdálenost a úhel. V tomto případě postačí měření vzdálenosti. Když je TCPF nastavený, musí se zařídit jeho stejná orientace s tou, kterou má fyzický robot. Pokud je pohled nastaven tak, že je vidět nápis SCHUNK, potom kladný směr osy Z je dolů, kladný směr pro Y je vlevo a kladný směr X je směrem k pozorovateli. Pokud by TCPF zůstal špatně natočený, potom při spojení s robotem dojde k přenastavení TCPF podle gripperu. Ale problém nastane v momentě, kdy přijde na definici Utool. Ten potom nebude souhlasit s tím, co je

nastaveno ve fyzickém robotovi a vygenerovaný program se nebude chovat tak, jak je požadováno. Proto je velice důležité nezapomenout otočit TCPF tak, jak bylo



Obr. 15 Gripper po definici TCPF

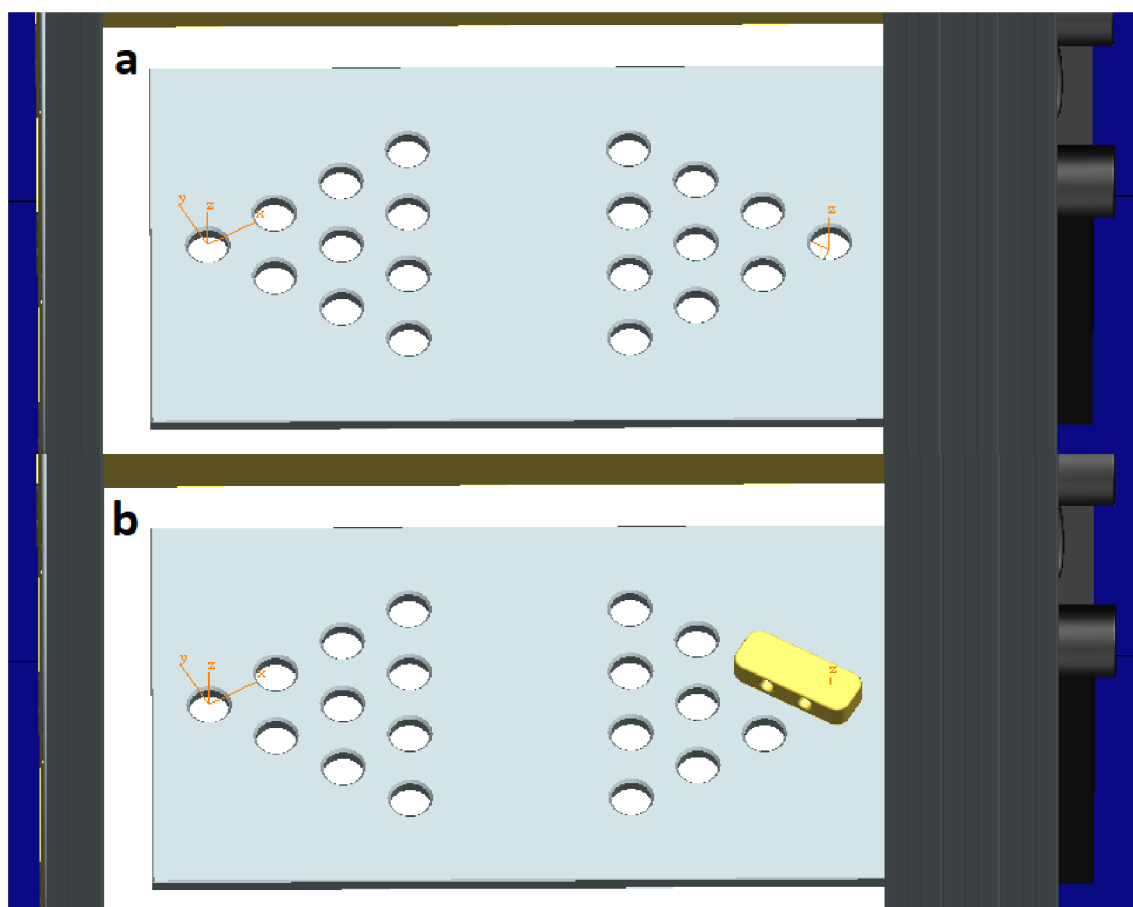
řečeno výše.

## 6.5 Robot v buňce a jeho simulace

Za pomoci předešlého cvičení byl gripper spojen s robotem a uveden do pohybu. Prvním krokem je vložení buňky, které se nastaví typ Work\_Table, následně je objekt otočen, aby měl souhlasný směr jako robot. Daný model se do programu vkládal jako typ Part.

Všechny dodané modely měly vhodně vytvořený souřadnicový systém, pomocí něhož bylo snadné součástku přesunout na požadovanou pozici. Polohu pro součástku, do níž by se mohla přemístit bylo nutné vytvořit příkazem Frame by 3 points, zde první zvolený bod je místo, kam se bude objekt přesouvat, druhý bod značí směr osy X. Pokud se stalo například, že osa Z směřovala špatným směrem, před potvrzením dialogu je zde možnost prohození, díky tomu osa Z následně směřuje správně. Kdyby osa zůstala směřovat špatným směrem a došlo k posunu objektu do tohoto středu, model by se objevil z druhé strany. Tento postup je nutno aplikovat jak na místo, kde je v úmyslu model zvednout, tak tam, kde má být položen. V případě této úlohy je velice užitečné posunout nově vytvořený souřadnicový systém, který je určen pro zvednutí modelu blíže k objektu s dírami. Tato hodnota se dá zjistit změřením vzdálenosti v programu NX. Jedná se o vzdálenost od plochy objektu, která se bude dotýkat objektu s dírami, k nejvyšší ploše vysunutí válce,

který má do daných děr zapadnout. Když je vše hotovo, pro přemístění dojde k použití Relocate.



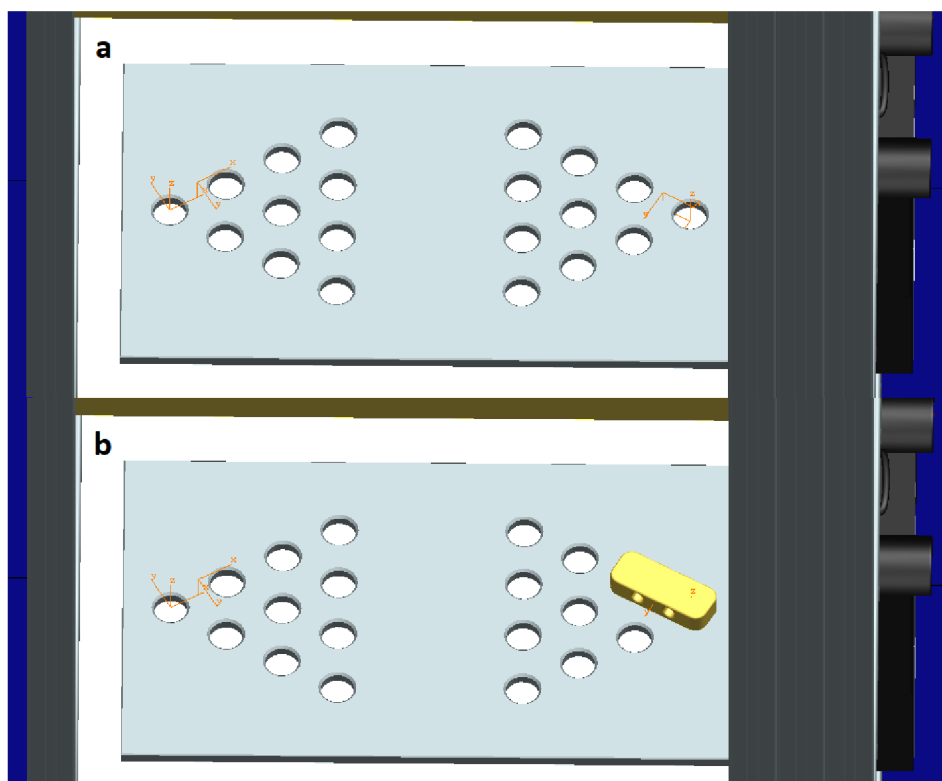
Obr. 16 Posunutý model na první místo

Následuje zjištění bodu, kam se má robot posunout a sevřít čelistmi daný model. Zde je velice vhodné zvolit metodu Frame between 2 points. Dané dva body je výhodné zvolit pro modely 1 a 2 na spodní straně, a pro modely 3 a 4 na vrchní straně. Pokud se pozorovatel pozorně podívá na modely, je z nich jasné, že měřit z druhé strany by byl zbytečně složitý krok. Jediný model číslo 1 tuhle větu popírá, díky jeho stavbě je jedno, ze které strany dojde ke měření vzdálenosti. Když jsou hodnoty změřeny, stačí přičíst nebo odečíst od jedné z koordinát XYZ pro oba body. Zde by nemusel být problém říct, že k posunu má dojít v ose Z. Avšak je možné, že ten kdo dělá tuhle úlohu z nějakého důvodu udělal takovou chybu, že si otočil všechny komponenty jiným směrem, než je v zadání a následně by byl tento člověk nucen posunout hodnotu v jiné než Z ose. Stejný postup se poté aplikuje na pozici, kde robot pustí model na koncové místo. Tedy za pomoci Relocate dojde k přesunu modelu a opakuje se stejný proces s metodou Frame between 2 points.

Naneštěstí body se s největší pravděpodobností vytvořily se špatnou orientací. Nápravu lze udělat za použití Placement Manipulator, pro tuto danou úlohu postačí



otáčení o úhly 90, -90 nebo 180 stupňů. Správná orientace je tak, aby kladný směr osy Z směřoval do objektu s dírami. Osa X musí směřovat stejným směrem, jako směřuje souřadnicový systém, který byl využit pro posun modelu do daného místa. Druhá možnost je, že by směřoval opačným směrem, tedy otočený o 180 stupňů vůči souřadnicovému systému, který byl využit pro posun modelu do daného místa. Na směr osy X nezáleží, protože robot si s tímto problémem jednoduše poradí díky možnosti otočit poslední šestý stupeň o dostatečnou hodnotu úhlu.



**Obr. 17** Souřadnicový systém pro Pick and Place operaci

Když jsou vytvořeny oba body a model je zpět na svém místě je možné vytvořit operaci Pick and Place. Zde se přiřadí robot, pro kterého se má tato operace vytvořit, s volbou robota je většinou spojen výběr gripperu pro přesun. Pokud tomu tak není je nutné ho zvolit také. Je možné, že jedním z důvodů, proč nebyl gripper automaticky vybrán je to, že není spojen s robotem. Po výběru gripperu se automaticky vyplní, co má gripper udělat, když je na pozici vyzvednutí nebo položení. Pro zvednutí by správně měla být pozice CLOSE a pro položení pozice OPEN. Dále se zde volí místo pro zvednutí a položení součástky. Tuto operaci bylo dále nutno doplnit o další kroky, ale prozatím postačí to, co je vytvořeno do teď. Protože nejdříve je nutné zjistit, jestli bylo vše nastaveno správně.



New Pick and Place Operation ✕

Name:

Robot:

Gripper:


Scope:


Gripper Pick and Place Poses:

Pick:

Place:

Define Pick and Place Points:

Pick:  

Place:  

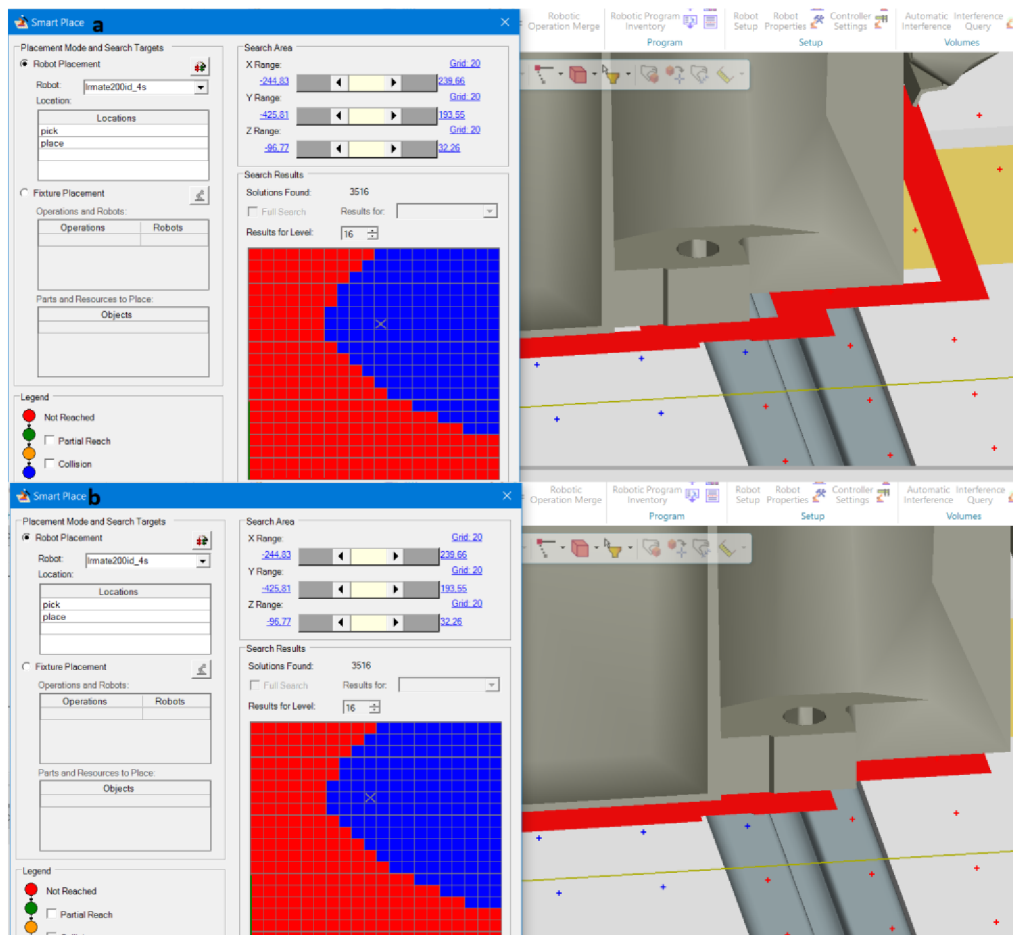
Use Existing Path:

Path:

Use Grip Frame

**Obr. 18** Nastavení operace Pick and Place

Při práci pro nějakého zákazníka by bylo vhodnější nejdříve udělat všechny pozice, do kterých je požadováno, aby se robot dostal a poté pokračovat v práci s funkcí Smart Place. Aby studenti neztráceli tolik času vytvářením programu, který by nemusel fungovat, protože by správně neseděly osy gripperu na robotovi a osy, do kterých má najet a zvednout model, je zvolena tato varianta. Tedy student si ověří správnost a následně může dál pokračovat. Kdyby měl student například špatně otočenou osu Z u části Pic, Place anebo u obou, funkce Smart Place by nenašla žádnou polohu, do které by se robot mohl posunout a fungovat. Funkce Smart Place funguje tak, že se do ní vloží všechny body, na které má daný robot dosáhnout. Dále se zvolí přibližné místo, kde by měl robot být postavený, a zvolí se kolik bodů se má v každé ose generovat. Funkce ukáže všechny pozice, na kterých robot dosáhne na vybrané operace, následně může dojít k přesunu na tyto pozice. Funkce Smart Place neslouží pro přesné usazení robota. Přesnou pozici je nutno doladit ručně.

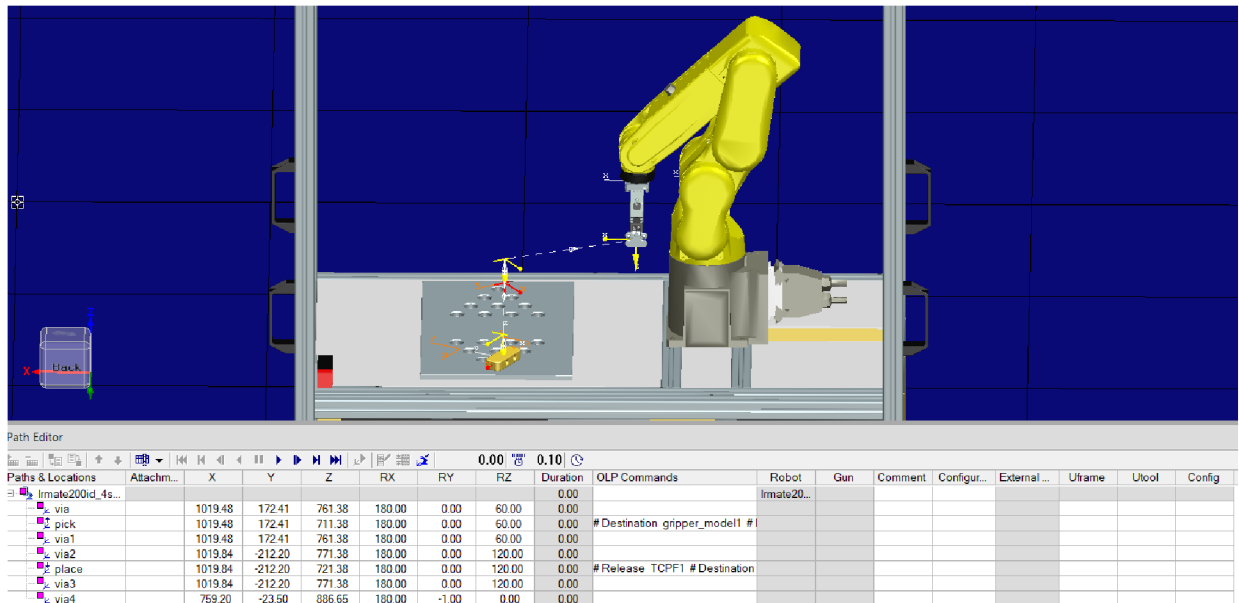


**Obr. 19 Usazení robota funkcí Smart Place**

Ruční doladění pozice robota může být mnohdy dosti náročná práce. Model buňky byl nejdříve upraven do podoby, aby bylo jasné, kam je nutné robota posunout. Aby studenti neměli velký problém s posunutím robota, byl ve středu objektu, na který se má robot posunout vytvořen souřadnicový systém. Ten normálně není vidět, k vidění je jenom, když je na buňku použit příkaz Set Modeling Scope. Tento obrázek je možné vidět na straně 19.

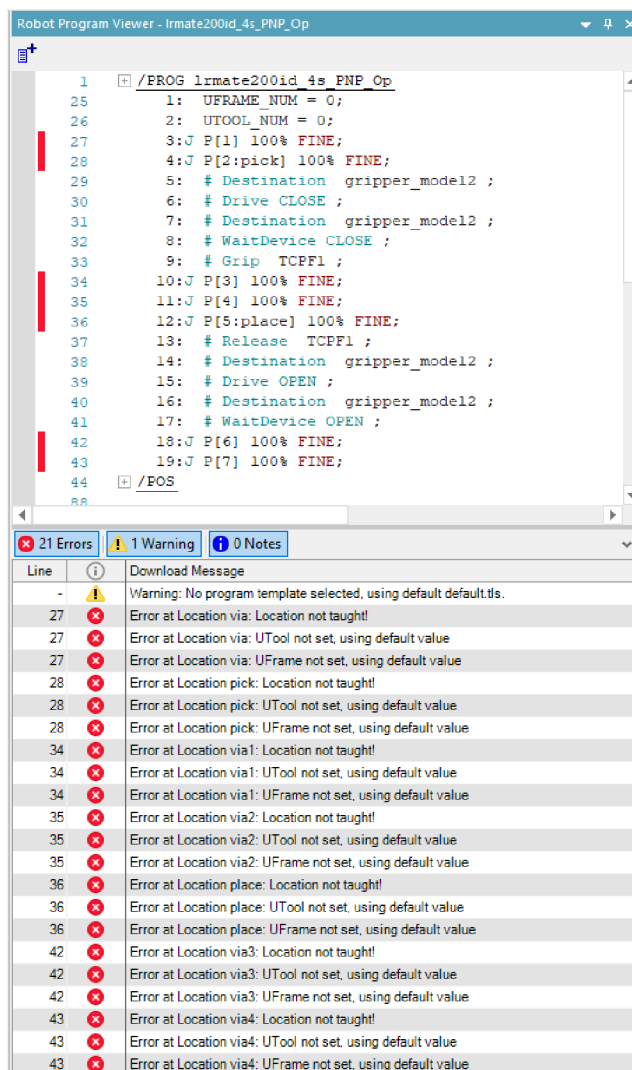
Když je robot na pozici a dosáhne všude kam má, nic nebrání tomu, aby program byl upraven do stavu, ve kterém by mohl fungovat bez možnosti kolize s dalšími předměty. Z tohoto důvodu se na první pozici dává krok, ve které je robot nad předmětem, který chce vzít. Dalším krokem se sjede níže na přesnou polohu a sevrou se kleště. Když je model sevřen, zvedne se robot do takové výšky, kde nehrozí kolize s dalšími modely. Stejný postup je aplikován na druhé straně. Posledním krokem je obvykle posunutí robota do domovské pozice. Pozorný student by mohl namítnout, proč není tahle domovská pozice přiřazena již na začátku celého programu. Robot, který má škola k dispozici nemá ve svém programu

pozici, kterou je možno vidět v TPS, z tohoto důvodu robot bude vždy startovat ze své domovské pozice, a ne ze stejné, jako je v TPS. Není problém, že robot vyjede z jiné pozice, protože je jeho úkolem dojet na danou pozici a pozice, ve které je právě teď nás nutně nezajímá. Nicméně je zvykem, aby robot startoval z pozice, ve které je například v TPS. Nyní by se mohlo zdát, že je vše hotovo. Pokud má člověk v úmyslu dělat pouze nástin dané operace, kudy se robot bude pohybovat a zda není špatný návrh kleští, potom se dá říct, že daná práce zde končí.



**Obr. 20 Kompletní virtuální simulace přesunu modelu**

Výhodou je pracovat s TPS ve verzi 14.1, v této verzi je možnost podívat se na program, popřípadě chyby, které jsou v daném programu. Je zde vidět, že není definovaný Uframe, Utool a robot nezná trasu. První věcí je zvolit správný kontrolér k danému robotovi. Kontrolér pro robota, který je k dispozici, je Fanuc-Rj. V nastavení robota se potom přiřadí defaultní Template, aby program nepsal varovné hlášení.



Obr. 21 Chyby viditelné v Robot Program Viewer

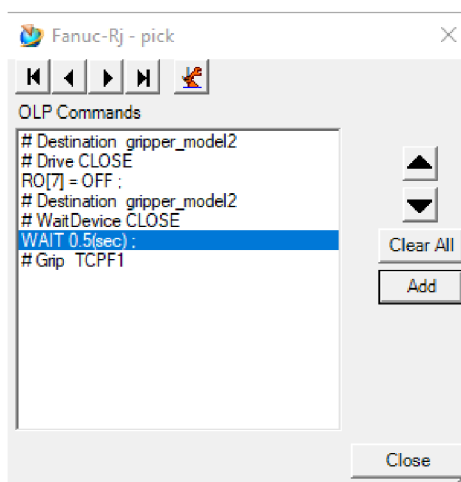
Další velice důležitou částí je definice Uframe a Utool. Je to z důvodu, že program TPS potřebuje vědět, jak má přepočítat dané pozice na to, co se dá dále použít pro fyzického robota. Je možnost definovat tento program od světového souřadnicového systému. Ten se nachází uprostřed druhého joint kloubu. Problém při téhle definici je to, že pokud dojde ke změně například náklonu podložky směrem k robotovi, změní se i souřadnice a program nebude fungovat. Avšak pokud použijeme nějaký nově nadefinovaný Uframe, který bude stejný jako v robotovi, a následně dojde k přesunu podložky, stačí přeučit robota Uframe a ten, jelikož má tyhle hodnoty svázané s daným Uframe, je přepočítá a program může dále fungovat. Definice Utool je velice jednoduchá, pokud je vytvořen TCPF gripperu, to z důvodu, že stačí vybrat TCPF a Utool je definován. Pro definici Uframe je potřeba mít tři souřadnicové systémy. Tím je zařízeno, že Uframe fyzického robota a toho v TPS je jeden a ten samý. V TPS postačí jeden souřadnicový systém, který je použit jako bod pro vyzvednutí modelu, ten se následně zkopíruje a dvakrát vloží. Jednomu je

přiřazeno jméno Uframe\_X a je posunut po záporných hodnotách osy X blíže směrem k robotovi, další Uframe\_Y je posunut po kladných hodnotách osy Y. Nejmenší vzdálenost, o kterou je možné posunout je udávána dva až tři palce. Když je Uframe a Utool definovaný je nutné ho přiřadit ke každému bodu v programu.

Paths & Locations	Attach...	X	Y	Z	RX	RY	RZ	Duration	OLP Commands	Robot	Gun	Comment	Configur...	External...	Uframe	Utool
lrmate200id_4s...								3.45		lrmate20...						
via		1019.48	172.41	761.38	180.00	0.00	60.00	0.73							4	2
pick		1019.48	172.41	711.38	180.00	0.00	60.00	0.27	#Destination gripper_model1 #						4	2
via1		1019.48	172.41	761.38	180.00	0.00	60.00	0.27							4	2
via2		1019.84	-212.20	771.38	180.00	0.00	120.00	0.70							4	2
place		1019.84	-212.20	721.38	180.00	0.00	120.00	0.42	#Release TCPF1 # Destination						4	2
via3		1019.84	-212.20	771.38	180.00	0.00	120.00	0.26							4	2
via4		1103.65	-23.50	1300.10	180.00	-90.00	0.00	0.80							4	2

Obr. 22 Definice Uframe a Utool

Další nedílnou součástí je nadefinovat povely pro otevření či zavření gripperu. To se nachází v sekci OLP Command. To nám otevře sekci, kde je možné přidávat mnoho dalších věcí na základě zvoleného kontroléru. Pro každého robota se tyto možnosti budou lišit. Studenti si vystačí s povely otevření nebo zavření kleští a času čekání. Čekání je velice důležité, protože pokud se robot pokusí odjet dříve, než sevře objekt mohlo by dojít ke špatnému sevření. To může mít za příčinu nezvednutí modelu, popřípadě jeho pád na další možné modely. V praxi by to mohlo znamenat škodu za nemalé peníze.

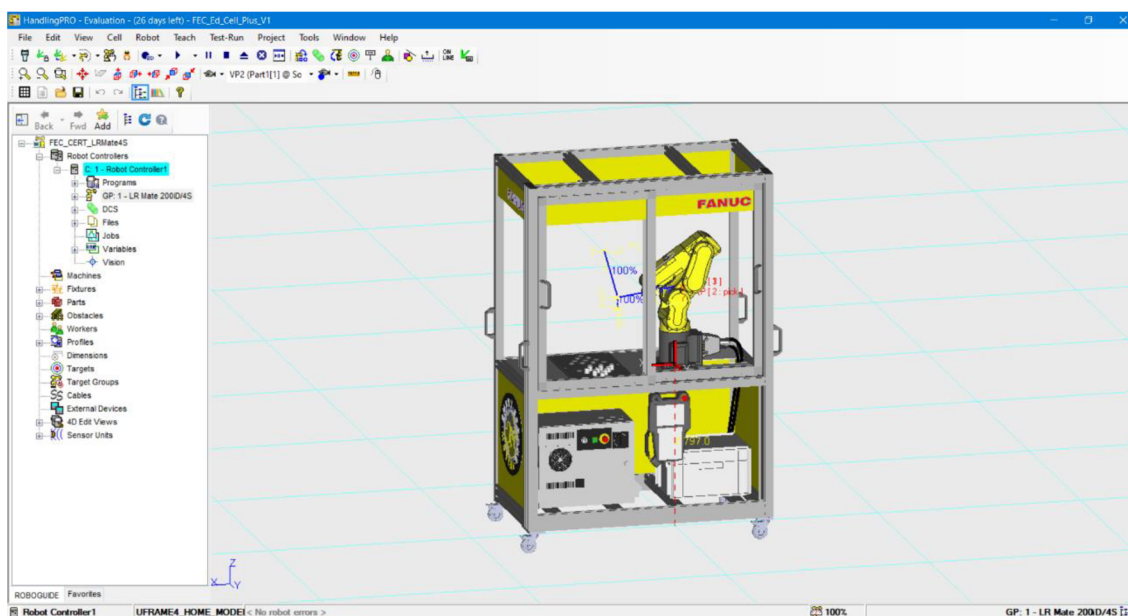


Obr. 23 Definování signálů pro robota

Nyní když je použito na robota Auto teach a je přehrána sekvence všech stavů, do kterých se má robot dostat, dojde k přepočtu hodnot od Uframe. Pokud by nyní došlo k otevření programu za pomoci Robot Program Viewer, neměla by se zde nacházet žádná chybová hláška. Pokud tomu tak je, může dojít k uložení programu do počítače, nabízené formáty jsou buď ls nebo pe.

Tento formát bohužel stále není možné nahrát do robota. Pokud by robot disponoval možností nahrát ASCII soubory, která by zařídila převod ASCII souboru

na binární, potom je možné za pomoci USB nahrát tento soubor do robota. Tam je převeden a je možné jej spustit. Další možností je použít program ROBOGUIDE. Využívá se k tomu HandlingPRO, zde je možno vybrat robota, pro který je nutné tento program převést. V tomto případě bylo využito toho, že byl k dispozici kompletní virtuální model buňky s daným robotem. Následně se nahraje vygenerovaný program, ROBOGUIDE přeloží daný program z ASCII do binárního typu. Po načtení programu je vidět cesta robota body, kterými má projet. Pokud jsou body svázané k nějakému Uframe, potom v programu ROBOGUIDE je cesta na špatném místě. Daný program počítá všechny body od jeho základního, takzvaného světového souřadnicového systému. Následně je možné program uložit ve formátu tp. V tomto formátu je program připravený na přesun za pomoci USB do fyzického robota. Pro přenos souborů se využívá teach pendant, volí se zde, do kterého USB je daný flash disk zapojen. Následně je zvolen typ souboru, který se má nahrát do robota, po této volbě jsou zobrazeny všechny soubory s touto koncovkou, posledním krokem je zvolit, který soubor má být převeden.



**Obr. 24 Buňka v programu ROBOGUIDE**

Jak již bylo zmíněno vygenerovaný program, který je spuštěn má chybu pokládací pozice. Dalším problémem je, že při nastavení Uframe na fyzickém robotovi je zde odchylka vůči programu 6,6 milimetrů v ose Z. Pro kontrolu byl Uframe nastaven i na druhém robotovi, který je k dispozici a zde byla odchylka 4,9 milimetrů. Vzhledem k navrženým čelistím jde vytvořit maximální rozdíl 1 milimetru.

## 7 ZÁVĚR

V první části této práce je krátce rozebrán vývoj průmyslu, které vynálezy způsobily největší posun dané doby. Dále jsou zde rozebrány některé důležité pojmy a principy, na které je možno narazit, pokud se daná osoba zajímá o průmysl 4.0. Nakonec je zde popsána chytrá továrna a přibližný funkční koncept, jak by taková továrna mohla asi fungovat.

Další kapitola se zabývá fyzickou buňkou od firmy FANUC. Pro koho je tato buňka vhodná, co od ní čekat. Pokud je ze stránek výrobce stáhnut manuál na ovládání buňky, je s trochou snahy možné, aby ji ovládal skoro každý. V závěru je krátce popsán program ROBOGUIDE a k čemu byl využit.

Další kapitola je věnovaná programu NX, který je návrhový 3D program, za jehož pomoci byla v semestrální práci převedena fyzická buňka na své digitální dvojče. Kvůli problémům zjištěným při další práci s touto buňkou byla doplněna o obrys robota. Je to z důvodu, že při následném vložení robota nebylo jasné, jestli je nebo není na správném místě. Pro spojení robota a gripperu od firmy SCHUNK byl vytvořen model příruby, který se shoduje s jeho fyzickou verzí na robotovi. Dále byly navrženy čelisti pro všechny čtyři modely. V jednom ze cvičení mají studenti za úkol navrhnout nějaké vlastní čelisti, je ale možné, že jejich návrh bude chybný, popřípadě zbytečně složitý na další možný 3D tisk. Z tohoto důvodu budou mít k dispozici čelisti, které byly pro tento účel navrženy.

Dále jsou popsány programy Mechatronic Concept Designer a Tecnomatix Process Simulate. Jsou popsány jejich výhody, co umí a k čemu jsou dobré. Následně je rozebráno, proč došlo k výběru programu TPS a ne MCD.

Hlavní částí této práce bylo navrhnout několik laboratorních úloh. První laboratorní úloha je vytvoření hliníkového profilu v programu NX. Následně vytvoří více těchto profilů a poskládají z nich rám fyzické buňky. Tento rám neodpovídá přesně tomu, jak vypadá fyzický model, došlo k menším úpravám pro zjednodušení práce.

Další vytvořenou úlohou je udělat program pro přesun modelu za pomoci ovládací komponenty teach pendant. V této úloze jsou studentům popsána všechna důležitá tlačítka na tomto ovladači. Dále se naučí vytvořit Utool a Uframe a poté si vytvoří kompletní program pro přesun.

Jelikož do daného cvičení byla vytvořena část laboratorní úlohy, na kterou bylo třeba navázat, bylo nutné zde pár věcí upravit. Důvodem bylo to, že došlo ke změnám v několika věcech, které byly ve starém zadání přepsány.

Následně byla vytvořena navazující část zadání, která úzce navazuje na předešlé cvičení. Zde studenti vloží virtuální model buňky s modelem, který má být přemístěn a následně vytvoří simulaci pro přesun. Simulace je výhodná pro ukázkou



zákazníkovi, jak bude co probíhat. Dále udělají věci potřebné, aby bylo možné program nahrát do robota. Tedy nadefinují Utool a Uframe, vytvoří signály pro otevření a zavření kleští a naučí robota dané body. Poté je program připraven pro export, kde musí být následně převeden na formát, se kterým dokáže robot pracovat a pak může být vyzkoušen na fyzickém robotovi.

Byla vytvořena simulace pro všechny modely, z této simulace je jako výstup vygenerovaný program pro každou součástku. V program je chyba pokládané pozice, která činí jeden milimetr. Pokud dojde k opravě děrované desky, měl by následně vytvořený program najet do správné pozice. Při pokusu vytvořit domovskou pozici v programu TPS došlo k otočení jointu j1, když nemělo. Dále je nutno vyřešit rozdíl v ose Z, tento a předešlý problém by mohl být způsoben chybou původní kalibrace. Chyba způsobená špatným návrhem desky nemá žádný vliv na vytvořenou laboratorní úlohu.



# Literatura

- [1] *Nová průmyslová revoluce – Průmysl 4.0* [online]. Olomouc: Szydlovská, 2017 [cit. 2019-12-21]. Dostupné z: <https://www.datamix.eu/blog/nova-prumyslova-revoluce-prumysl-4-0/>
- [2] *Průmysl 4.0* [online]. 1 548 Antonínská Brno-střed Brno Česká republika, 2019 [cit. 2019-12-21]. Dostupné z: [https://www.vutbr.cz/www\\_base/priloha\\_fs.php?dpid=185233&skupina=dokument\\_priloha](https://www.vutbr.cz/www_base/priloha_fs.php?dpid=185233&skupina=dokument_priloha). Učební text VUT, prezentace. Vysoké učení technické v Brně. Vedoucí práce Ing. Václav Kaczmarczyk, Ph.D.
- [3] *Industry 4.0* [online]. Děčín: Automa, 2015 [cit. 2019-12-21]. Dostupné z: [http://www.automa.cz/Aton/FileRepository/pdf\\_articles/54254.pdf](http://www.automa.cz/Aton/FileRepository/pdf_articles/54254.pdf)
- [4] *Školení v obsluze robotů pro školy a univerzity* [online]. [cit. 2019-12-22]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/robotick%C3%A9-%C5%99e%C5%A1en%C3%AD-pro-%C5%A1koly>
- [5] *Inteligentní 3D simulace robotu* [online]. [cit. 2019-12-22]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/p%C5%99%C3%ADslu%C5%A1enstv%C3%AD/robotguide>
- [6] *Products for Digital Innovation* [online]. [cit. 2019-12-23]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/products/>
- [7] *NX CAD* [online]. [cit. 2019-12-23]. Dostupné z: <https://www.axiomtech.cz/24790-cad-nx>
- [8] *NX CAD* [online]. [cit. 2019-12-23]. Dostupné z: <https://www.tpvgroup.cz/nx-cad-110/>
- [9] NX (program). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-12-23]. Dostupné z: [https://cs.wikipedia.org/wiki/NX\\_\(program\)](https://cs.wikipedia.org/wiki/NX_(program))
- [10] MCD\_Quick\_Start. *MCD\_Quick\_Start* [online]. Los Angeles, 2010, , 5-10 [cit. 2020-06-02]. Dostupné z: [http://alvarestech.com/temp/PDP2011/SiemensNXMechatronicsConceptDesigner/MCD\\_Quick\\_Start.pdf](http://alvarestech.com/temp/PDP2011/SiemensNXMechatronicsConceptDesigner/MCD_Quick_Start.pdf)
- [11] *Mechatronic Concept Designer* [online]. [cit. 2020-06-02]. Dostupné z: <https://www.axiomtech.cz/25312-mechatronic-concept-designer>
- [12] *Process Simulate Human* [online]. [cit. 2019-12-28]. Dostupné z: <https://www.axiomtech.cz/25345-process-simulate-human>

[13] *Process Simulate Robotics* [online]. [cit. 2019-12-28]. Dostupné z:  
<https://www.axiomtech.cz/25343-tecnomatix-process-simulate-robotics>

## Seznam symbolů, veličin a zkratk

TPS	...	Tecnomatix Process Simulate
MCD	...	Mechatronic Concept Designer
TCPF	...	Relativní střed pracovního nástroje

# Seznam příloh

Příloha 1 – Obsah přiloženého CD .....	39
--	----

## Příloha 1 – Obsah přiloženého CD

Přiložené CD obsahuje:

- Elektronickou podobu bakalářské práce
  - o Ve složce *Bakalářská práce*
- Elektronickou podobu návodu do laboratorních cvičení
  - o Ve složce *Laboratorní cvičení*
- Simulaci pro model 1
  - o Ve složce *Model1*
  - o Ve složce chybí robotické rameno, protože podléhá licenci
- Simulaci pro model2
  - o Ve složce *Model2*
  - o Ve složce chybí robotické rameno, protože podléhá licenci
- Simulaci pro model3
  - o Ve složce *Model3*
  - o Ve složce chybí robotické rameno, protože podléhá licenci
- Simulaci pro model4
  - o Ve složce *Model4*
  - o Ve složce chybí robotické rameno, protože podléhá licenci
- Program pro model1 ve formátu ls a tp
  - o Ve složce *Program Model1*
- Program pro model2 ve formátu ls a tp
  - o Ve složce *Program Model2*
- Program pro model3 ve formátu ls a tp
  - o Ve složce *Program Model3*
- Program pro model4 ve formátu ls a tp
  - o Ve složce *Program Model4*