



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH ŘÍDÍCÍHO PROGRAMU PRŮMYSLOVÉHO
ROBOTA IRB 120 POMOCÍ SOFTWARE ROBOTSTUDIO**

DESIGN OF THE CONTROL PROGRAM FOR INDUSTRIAL ROBOT IRB 120 USING ROBOTSTUDIO
SOFTWARE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Václav Nevřiva

VEDOUcí PRÁCE

SUPERVISOR

Ing. Roman Parák

BRNO 2019

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Václav Nevřiva**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. Roman Parák**
Akademický rok: 2018/19

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

Návrh řídicího programu průmyslového robota IRB 120 pomocí softwaru RobotStudio

Stručná charakteristika problematiky úkolu:

Práce bude zahrnovat rešerši průmyslového robota IRB 120 a seznámení se řídicími prostředky společnosti ABB Robotics a vývojovým prostředím RobotStudio.

Předmětem práce bude návrh řídicího programu pro vybranou laboratorní úlohu. Následovat bude implementace navrženého řešení. Závěr práce bude věnován ověření navrženého řešení simulací a následně na reálném hardwaru.

Práce předpokládá aktivní přístup studenta a nutnost práce v laboratoři.

Cíle bakalářské práce:

Seznamte se s řídicími prostředky společnosti ABB Robotics a vývojovým prostředím RobotStudio.

Proveďte stručnou rešerši průmyslového robota IRB 120.

Navrhněte řídicí program pro vybranou laboratorní úlohu.

Implementujte návrh na řídicích prostředcích ABB Robotics.

Ověřte funkčnost vytvořeného řešení.

Seznam doporučené literatury:

ABB Robotics: Manufacturer & Supplier of Industrial Robots.<https://new.abb.com/products/robotics>

IRB 120: Industrial Robots (Robotics).<https://new.abb.com/products/robotics/industrial-robots/irb-120>

KOLÍBAL, Zdeněk. Roboty a robotizované výrobní technologie. Brno: Vysoké učení technické v Brně - nakladatelství VUTUM, 2016. ISBN 978-80-214-4828-5.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Cílem bakalářské práce je návrh řídicího programu pro vybranou laboratorní úlohu v programu RobotStudio, jeho implementace pro průmyslového robota IRB 120 společnosti ABB Robotics a ověření navrženého řešení simulací a testem na reálném hardware. V úvodní části je stručně představena společnost ABB, produkty společnosti a vývojové prostředí RobotStudio. Následující kapitoly popisují vybranou laboratorní úlohu, její řešení a testování navrženého programu. Bakalářská práce byla rozšířena o návrh uživatelského rozhraní pro ovládání úlohy a ukázkou práce s robotem s využitím virtuální reality.

ABSTRACT

The aim of the bachelor thesis is to design a control program for a selected laboratory task in RobotStudio program, its implementation for the industrial robot IRB 120 from ABB Robotics and to verify the proposed solution by simulation and test on real hardware. The introductory part introduces ABB, company products and the development environment. The following chapters describe the selected laboratory task, its solution and testing of the proposed program. The bachelor thesis was extended with the design of user interface for controlling the task and demonstration of working with the robot using virtual reality.

KLÍČOVÁ SLOVA

Hanojské věže, ABB Robotics, IRB 120, RobotStudio, ScreenMaker, FlexPendant, virtuální realita

KEYWORDS

Tower of Hanoi, ABB Robotics, IRB 120, RobotStudio, ScreenMaker, FlexPendant, virtual reality

BIBLIOGRAFICKÁ CITACE

NEVŘIVA, Václav. *Návrh řídicího programu průmyslového robota IRB 120 pomocí softwaru RobotStudio*, Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky. Vedoucí práce Roman Parák.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce Ing. Romanu Parákovi za odborné vedení, trpělivost a vstřícnost v průběhu řešení této práce. Dále bych rád poděkoval mé rodině za podporu a motivaci v průběhu celého studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Romana Paráka a s použitím literatury uvedené v seznamu literatury.

V Brně dne 24. 5. 2019

.....

Václav Nevřiva

OBSAH

1	ÚVOD.....	15
2	ABB ROBOTICS	17
2.1	O společnosti ABB	17
2.1.1	Divize ABB Robotics – robotika a automatizace	18
2.1.2	Roboty společnosti ABB	18
2.2	Robot IRB 120	22
2.2.1	Stavba robota	23
2.2.2	Řídící systémy robota	24
2.3	Vývojové prostředí RobotStudio	25
2.3.1	Objekty	26
2.3.2	Nástroj	27
2.3.3	Cesty, pohyby a targety	27
2.3.4	I/O Signály	28
2.3.5	Rapid.....	29
2.3.6	Simulace	30
2.3.7	ScreenMaker	30
3	VLASTNÍ ŘEŠENÍ ÚLOHY	33
3.1	Výběr laboratorní úlohy.....	33
3.1.1	Hanojské věže.....	33
3.1.2	Možnosti řešení.....	33
3.1.3	Volba použitého řešení	34
3.2	Řešení úlohy	35
3.2.1	Návrh prostředí úlohy	35
3.2.2	Návrh efektoru	36
3.2.3	Návrh řídicího programu	37
3.2.4	Tvorba uživatelského rozhraní	40
4	TESTOVÁNÍ.....	43
4.1	Simulace úlohy v programu RobotStudio	43
4.2	Ověření úlohy na reálném hardware	45
5	POUŽITÍ VR TECHNOLOGIÍ	47
5.1	Technologie virtuální reality	47
5.2	Ukázka propojení VR s programem RobotStudio	48
5.3	Možnosti dalšího vývoje a praktického použití	51
6	ZÁVĚR	53
7	POUŽITÉ INFORMAČNÍ ZDROJE	55
8	SEZNAM OBRÁZKŮ A TABULEK.....	59
8.1	Seznam tabulek.....	59
8.2	Seznam obrázků.....	59
9	SEZNAM PŘÍLOH.....	61

1 ÚVOD

Cílem této bakalářské práce je návrh řídicího programu pro vybranou laboratorní úlohu, logický hlavolam hanojské věže, v programu RobotStudio od společnosti ABB Robotics a jeho implementace pro průmyslového robota IRB 120. Dále otestování navrženého programu pomocí simulace a následně na reálném hardwaru v laboratoři.

V úvodní části je stručně představena divize společnosti ABB zabývající se robotikou, její produkty, řídicí systémy a roboty. Následně její vývojové prostředí pro návrh robotických pracovišť, program RobotStudio spolu s nástrojem ScreenMaker pro tvorbu grafického uživatelského rozhraní. V následující kapitole je popsána samotná laboratorní úloha, rozebrány možnosti jejího řešení, a z nastíněných možností je vybráno řešení, které bude použito pro tvorbu řídicího programu. Dále je v této kapitole popsán návrh výsledného řídicího programu a jeho implementace na produktech společnosti ABB. Kapitola je také doplněna o návrh grafického uživatelského rozhraní, pomocí kterého lze výsledný program snadno ovládat a podrobovat dalším testům. Další kapitola se věnuje problematice testování navrženého programu s využitím počítačové simulace, následně testům ve školní laboratoři na reálném hardware, a nakonec vyhodnocení a porovnání těchto testů. Poslední kapitola se věnuje problematice použití technologie virtuální reality při práci s výrobním robotem a při návrhu robotického pracoviště. Shrnuje současný stav poznání a popisuje jednoduchý ilustrační test, při kterém byly demonstrovány možnosti propojení aplikace využívající virtuální realitu a vývojového prostředí společnosti ABB. V závěru kapitoly jsou stručně nastíněny možnosti dalšího vývoje a praktického použití této technologie.

2 ABB ROBOTICS

2.1 O společnosti ABB

ABB (Asea Brown Boveri) je švédsko-švýcarská nadnárodní korporace se sídlem v Curychu, která vznikla fúzí společností ASEA a Brown, Boveri & Cie. S více než stotřicetiletou tradicí, zastoupením ve 102 zemích světa a takřka 148 000 zaměstnanci je jedním ze světových lídrů v oblastech energetiky, elektrotechniky, robotiky a automatizace. V roce 2018 společnost vykázala tržby ve výši 28,6 mld USD, což je více než 650 mld korun. V České republice formálně působí od roku 1992 a dnes má na našem území pět výzkumných center a šest výrobních závodů, které se nachází v Praze, Brně, Plzni, Ostravě, Trutnově a Jablonci nad Nisou, a v kterých společnost zaměstnává více než 3700 zaměstnanců.

Provozně se společnost dělí do pěti divizí, které kopírují její hlavní zájmová odvětví. Jedná se o divize energetiky a elektrotechnické výroby, které nabízející energetické výrobky a softwarová řešení pro výrobu, přenos a rozvod elektrické energie, divizi průmyslové automatizace zabývající se výrobky a službami pro optimalizaci produktivity průmyslových procesů, divizi pohonů zabývající se výrobou a vývojem elektromotorů a frekvenčních měničů a nakonec o divizi robotiky a automatizace. Dle informací oznámených na konci roku 2018 by měla být divize elektrotechnické výroby prodána a v roce 2020 převedena pod japonské konsorcium Hitachi. [1, 2]

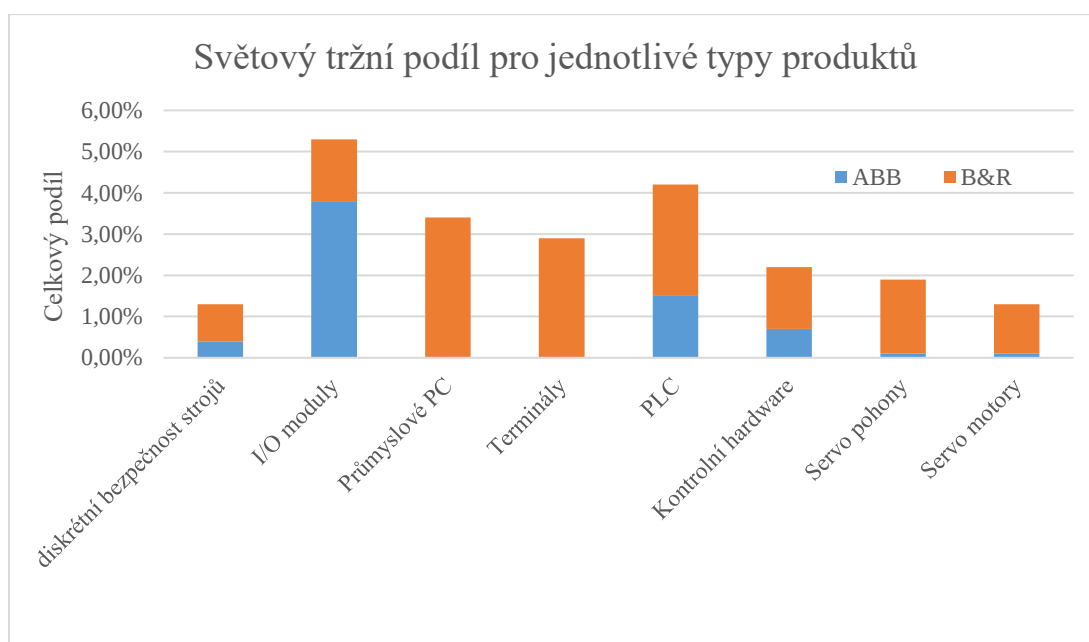


Obr. 1: Hlavní obchodní centrum ABB v Krakově, Polsko [3]

2.1.1 Divize ABB Robotics – robotika a automatizace

Divize robotiky a automatizace se zabývá výrobou a vývojem průmyslových robotů, řídicích systémů, aplikačního software a příslušenství. Do roku 2019 ABB nainstalovalo více než 300 000 robotů po celém světě. Divize robotiky a automatizace se také stará o vývoj a distribuci programu RobotStudio pro programování robotů offline, provozuje školicí centra a zajišťuje školení jak pro konstruktéry robotizovaných pracovišť, tak pro operátory jednotlivých průmyslových robotů. [4]

V roce 2017 provedla společnost ABB akvizici společnosti B&R (Bernecker + Rainer Industrie-Elektronik GmbH), která je největším nezávislým dodavatelem produktových a softwarových řešení s otevřenou architekturou pro automatizaci strojů a výrobních procesů. Tímto krokem došlo k zaplnění mezery v portfoliu ABB, která se těmto produktům historicky příliš nevěnovala. Software a řešení pro moderní automatizaci strojů a výrobních procesů doplňují nabídku společnosti ABB v oblasti robotů, procesní automatizace, digitalizace a energetiky. Centrála B&R v Rakousku se má nově také stát globálním centrem ABB pro automatizaci strojů a výrobních procesů. [5, 6]



Obr. 2: Společný tržní podíl po akvizici B&R ve vybraných odvětvích, přeloženo z anglického originálu [7]

2.1.2 Roboty společnosti ABB

Roboty v portfoliu firmy ABB můžeme rozdělit do čtyř kategorií, které se vzájemně liší stavbou a především účelem.

První a nejrozsáhlejší kategorií jsou roboty označené jako víceúčelové. Jedná se o roboty se sériovou kinematikou, na jejichž koncovou přírubu lze připevnit libovolný kompatibilní koncový efektor, jak z portfolia firmy, tak dle vlastního návrhu, díky čemuž

jsou možnosti jejich využití velmi široké. Dle použitého efektoru mohou být používány k manipulaci břemen, popisu dílů, svařování či balení produktů pohybujících se po dopravníku. V současné době se v nabídce společnosti nachází 21 typů víceúčelových robotů, lišících se především maximální nosností, která se pohybuje od 3 kg u nejmenšího typu, IRB 120, až po 800 kg u rozměrného typu IRB 8700 a velikostí manipulačního prostoru, která začíná na 45 centimetrech a končí až na dosahu 4,2 metru u největších manipulátorů. [8, 9]



Obr. 3: Univerzální robot IRB 8700 v porovnání s člověkem [10]

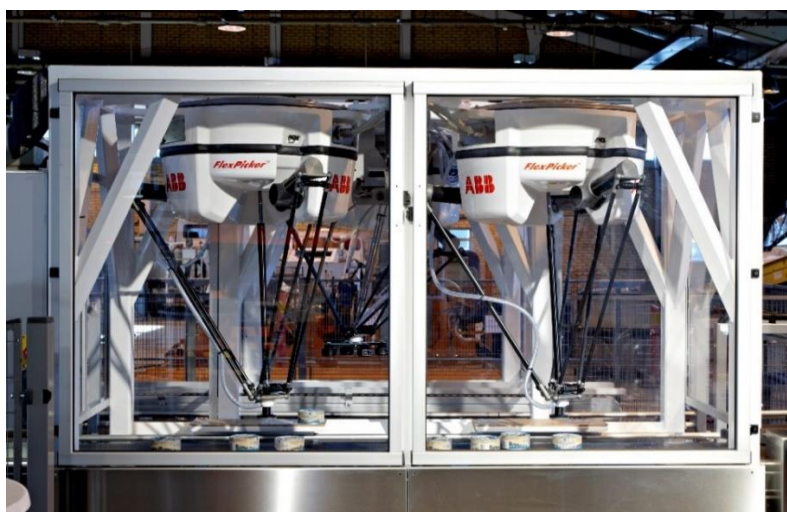
Druhou kategorií v portfoliu firmy jsou roboty kolaborativní. To znamená, že mohou při práci, na rozdíl od výše zmíněných robotů, kteří musí být bezpečně odděleni, spolupracovat přímo s člověkem bez rizika jeho poranění v případě nehody. Robot za pomoci senzorů neustále monitoruje svou rychlost a sílu kterou vynakládají jeho členy. V případě, že se sledované parametry odchýlí od maximálních stanovených hodnot pro zadanou úlohu, což by mohlo znamenat například kontakt robota s člověkem, robot ihned zastaví a zamezí tak možnému úrazu či jiné kolizi. Často jsou také tyto roboty instalované spolu se senzory a kamerami, které sníží jeho rychlost v případě, že se operátor dostane do jeho blízkosti, a tím ještě více minimalizují riziko nehody. Tato nová kategorie robotů, která se pravděpodobně bude v budoucnu dále rozvíjet přináší zcela nové možnosti využití robotů a potenciálně snižuje náklady na konstrukci robotizovaných pracovišť, které pak nemusí být vybaveny fyzickými zábranami, optickými závorami a dalším bezpečnostním vybavením. Jistou nevýhodou těchto robotů jsou nižší pracovní rychlosti a složitější konstrukce koncových efektorů, které musí být vybaveny pokročilejšími senzory působících sil a momentů. V současné době firma ABB nabízí dva podobné kolaborativní roboty, a to IRB 14000 YuMi, který je vybaven dvěma manipulačními rameny a jeho zjednodušenou variantou s jedním ramenem, IRB 14050 Single-arm

YuMi. Tyto kolaborativní roboty dokáží manipulovat s břemenem o hmotnosti až 0,5 kg v manipulačním prostoru o velikosti 559 mm. [9, 11, 12]



Obr. 4: Kolaborativní robot IRB 14000 YuMi [13]

Třetí kategorií jsou roboty rodiny IRB 360 FlexPicker s paralelní kinematikou, instalující se typicky nad dopravníkové pásy. Tyto roboty nabízejí ve srovnání s klasickou těžkou automatizací vysokou flexibilitu a rychlost na kompaktní ploše při současném zachování přesnosti a vysokém užitečném zatížení. Tyto roboty jsou často používány k manipulaci s drobnými díly, balení a třídění výrobků ve spolupráci s kamerami a programem pro detekci objektů v obraze. Roboty IRB 360 jsou vhodné i do podmínek s vysokými požadavky na hygienu, například v masném či mlékárenském průmyslu, a to díky stupni krytí IP69K, který zajišťuje omyvatelnost průmyslovými čistícími prostředky a vysokotlakou horkou vodou. Důležité také je, že klouby ramen robota nevyžadují maziva, což dále minimalizuje možnosti znečištění výrobků. Rodina robotů IRB 360 obsahuje 4 varianty s užitečným zatížením 1 kg, 3 kg, 6 kg a 8 kg s dosahem 800 mm, 1130 mm a 1600 mm. [9, 14, 15]



Obr. 5: Dvojice robotů IRB 360 FlexPicker balící konzervy, Jihoafrická Republika [16]

Čtvrtou kategorií jsou roboty rodiny IRB 910SC označované jako SCARA. Jedná se o jednoramenné roboty, vhodné k manipulačním úkonům s menšími výrobky v prostorově náročném prostředí. Používají se například pro kytování trayů, ukládání zboží na palety, vkládání a odběr materiálu ze strojů a další úkony, vyžadující rychlý, opakovatelný a obratný pohyb mezi dvěma místy. Roboty rodiny SCARA mají nosnost 6 kg a vyrábějí se ve třech modulárních provedeních o délce ramene 450 mm, 550 mm a 650 mm. [17]



Obr. 6: Robot IRB 910SC s délkou ramene 550 mm [18]

Poslední, pátou kategorií jsou lakovací roboty. Jedná se o jednoúčelné roboty určené k instalaci na lakovacích výrobních linkách, většinou v automobilovém průmyslu. Součástí robotů je sofistikovaný systém kontroly nanášení barvy a samotné rozprašovací zařízení. V této kategorii společnost ABB nabízí také jednoúčelné roboty k jednoduchým manipulačním úkonům na výrobní lince jako je otevírání dveří či kapot lakovaných karoserií. Jednotlivé roboty určené přímo k nanášení barvy se liší především svou velikostí a úměrně tomu velikostí pracovního prostoru, která se pohybuje od 1,2 do 4,1 metru. [19]



Obr. 7: Lakovací robot IRB 52 určený k lakování menších dílů na výrobních linkách [20]

2.2 Robot IRB 120

Univerzální průmyslový robot ABB IRB 120 je nejmenším robotem z portfolia firmy ABB. Jeho maximální užitečné zatížení je 3 kg a maximální dosah 580 mm. Robot je určen především pro manipulaci a přesnou montáž dílů. Hmotnost zařízení 25 kg a štíhlý design umožňují snadné umístění manipulátoru do stísněných výrobních buněk, na výrobní stroje nebo do blízkosti dalších robotů jako součást výrobní linky. IRB 120 nalézá uplatnění v rozsáhlém spektru průmyslových segmentů, jako je elektrotechnický průmysl, strojní výroba, potravinářství nebo také farmacie. Díky svým malým rozměrům a hmotnosti jej také můžeme často nalézt jako součást výzkumných testovacích stanic, či jako vhodného testovacího robota pro nová řešení, výzkum a testy v oblasti automatizace.



Obr. 8: Robot ABB IRB 120 na produktové fotografii [21]

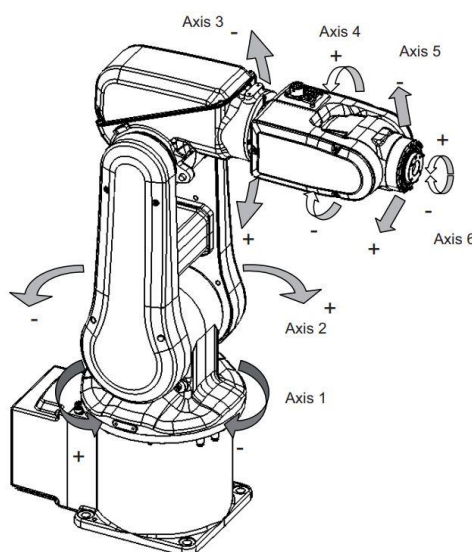
Robot ABB IRB 120 se vyrábí v několika variantách, a to jednak ve variantě pro inverzní montáž, kdy je robot zavěšen na své přírubě a jeho pracovní prostor se nachází pod ním, tak ve variantě pro čisté provozy podle ISO 5 s certifikací IPA označované jako Clean Room, kdy robot obsahuje maziva splňující technické a legislativní požadavky pro aplikaci v potravinářském průmyslu a zároveň má hladkou povrchovou úpravu a speciální nátěr, usnadňující jeho čištění. Novinkou je verze IRB 120T, zaměřená na aplikace „pick and place“, při kterých dochází k opakovanému úchopu a uložení výrobků, která nabízí rychlejší pohyb os oproti verzi základní, což vede ke snížení délky trvání pracovního cyklu až o 25 %. Srovnání maximálních rychlostí jednotlivých os je vidět v Tab.1. Osy jsou číslovány ve směru od základny.

Tab. 1: Srovnání rychlostí jednotlivých os robotů ABB IRB120 a ABB IRB 120T [22]

Typ robota	Osa 1	Osa 2	Osa 3	Osa 4	Osa 5	Osa 6
IRB 120	250°/s	250°/s	250°/s	320°/s	320°/s	420°/s
IRB 120T	250°/s	250°/s	250°/s	420°/s	590°/s	600°/s

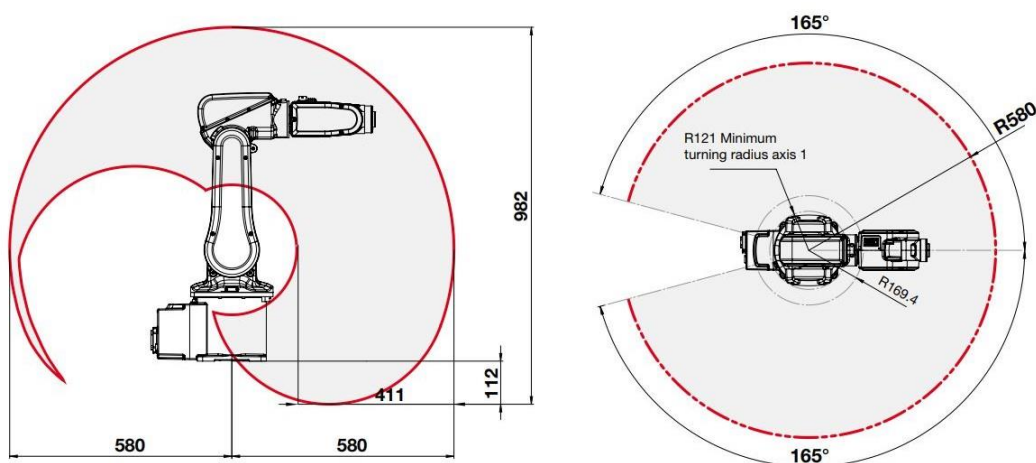
Dle ISO9283 robot dosahuje při maximálním zatížení a pohybu všech os přesnosti opakování pozice 0,01 mm a přesnosti dosažení požadované pozice 0,02 mm. Ustálení robota na pozici trvá do 0,03 sec. [22, 23, 24]

2.2.1 Stavba robota



Obr. 9: Robot IRB 120 s vyznačenými osami [23]

Výrobní robot ABB IRB 120 má šest pohyblivých os. Jednotlivé členy jsou řazeny sériově, jedná se tedy o robota s tzv. sériovou kinematikou. Schématický náčrtek jednotlivých os a náčrtek pracovního rozsahu robota lze vidět na Obr.9 a Obr.10. Základna robota má tvar čtverce o rozměrech 180x180 mm a je opatřena čtyřmi otvory o průměru 12 mm, sloužícími k upevnění robota na pracovní pozici pomocí šroubů. Koncová příruba k montáži efektoru je tvaru kruhu s průměrem 40 mm, přičemž k pevnému a přesnému uchycení nástroje jsou připraveny čtyři díry se závitem o průměru 5 mm.



Obr. 10: Náčrtek pracovního prostoru robota IRB 120 [23]

K ovládání efektoru jsou vnitřkem robota na horní rameno vyvedeny 4 vzduchové přívody, které mohou pracovat s tlakem až 5 barů, a konektor s 10 piny pro napájení nástroje nebo vedení signálů o maximálním napětí 49 V a proudu 500 mA. Vstupy

k těmto přívodům jsou umístěny na zadní části základny, vedle konektorů pro vedení, kterými je robot připojen k řídicí jednotce. [22, 23]

2.2.2 Řídicí systémy robota

V současné době ABB spolu s roboty dodává již pátou generaci řídicích systémů, označovanou jako IRC5. Řídicí jednotka se dodává ve dvou variantách, a to kromě základní varianty také v kompaktní variantě označované jako IRC5C. Kromě rozměrů a hmotnosti se liší především tím, že k řídicí jednotce klasické velikosti lze připojit až 3 tzv. MultiMove moduly, díky kterým lze ovládat více robotů, maximálně 4, z jediné řídicí jednotky. Klasická řídicí jednotka také splňuje vyšší úroveň ochrany proti vniknutí kapaliny a prachu, a to IP54 oproti IP20 u kompaktní verze.

Tab. 2: Srovnání parametrů řídicích jednotek IRC5 a IRC5C [25]

Typ řídicí jednotky	Rozměry	Hmotnost	Stupeň krytí	Rozsah pracovních teplot
IRC5	970x725x710 mm	150 kg	IP54	0-52 °C
IRC5C	310x449x442 mm	30 kg	IP20	0-45 °C

Řídicí jednotky používají technologii automatické optimalizace výkonu robota ke zkrácení času pracovního cyklu označovanou jako QuickMove a technologii přesného řízení pohybu po zadané trajektorii označovanou jako TrueMove. O bezpečnost operátora se stará systém SafeMove2, umožňující fungování systému s kolaborativními roboty. [25, 26]



Obr. 11: Řídicí jednotka IRC5C [25]

K řídicí jednotce je možno připojit přenosný ovladač, FlexPendant. Ten je vybaven barevným dotykovým displejem pro zobrazování informací o stavu robota, programování robota online a spouštění vlastních aplikací ke konkrétním úlohám. Dále je opatřen 3D joystickem pro přímé ovládání pohybu robota a základními tlačítky pro

spuštění/zastavení aktuálního programu, či přepínání souřadného systému. Bezpečnost zajišťuje spínač na rukojeti, bez jehož sepnutí nelze s robotem pohybovat, a tlačítko nouzového zastavení. [27]



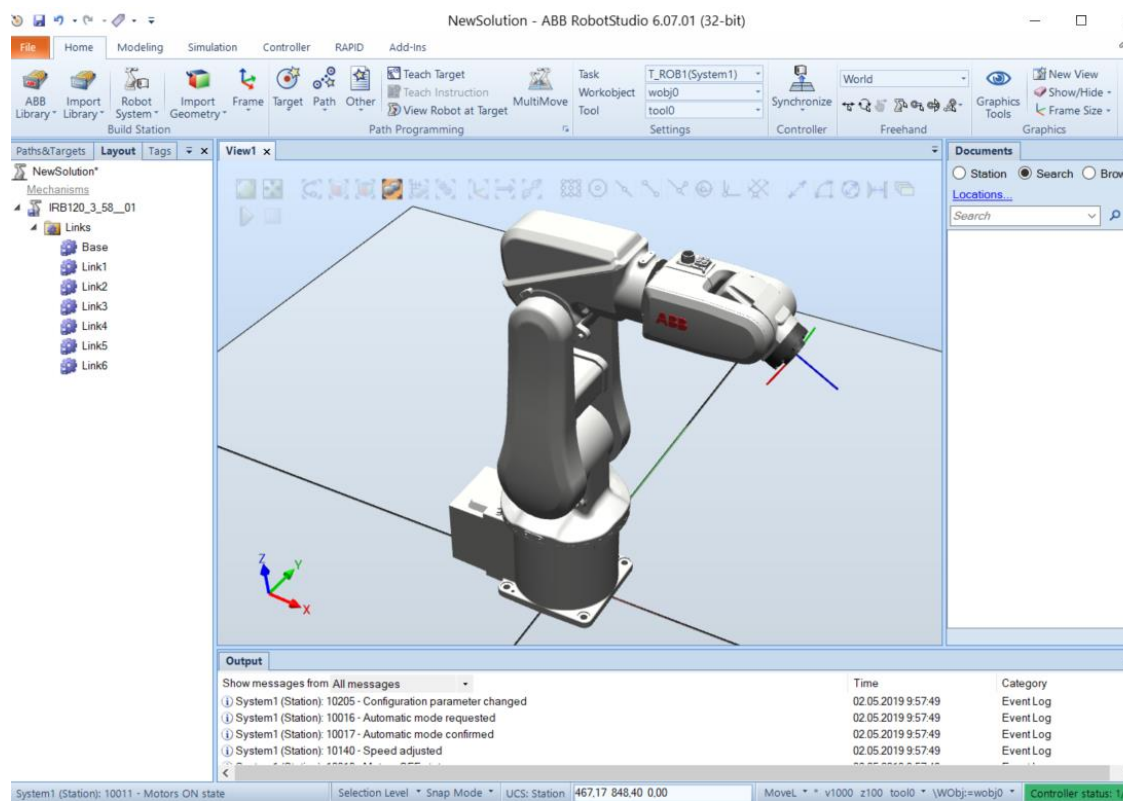
Obr. 12: FlexPendant [28]

2.3 Vývojové prostředí RobotStudio

RobotStudio je komplexní software pro programování robotů offline od společnosti ABB, umožňující simulaci řídicího systému, na něj napojeného robota nebo skupiny robotů i dalších prvků prostředí, jako je vybavení robotické pracovní stanice, či výrobky se kterými robot pracuje. Program umožňuje běh virtuální řídicí jednotky, tzv. virtual controlleru, která je kopií skutečného řídicího systému a spolu s propracovanou fyzikální simulací samotného robota umožňuje přesnou vizualizaci a kontrolu řešených úloh. Jako virtuální řídicí jednotku lze použít již existující řídicí systém se všemi v minulosti importovanými programy a signály, což umožňuje jednoduše a spolehlivě upravovat a vylepšovat již existující robotizované pracoviště. [29]

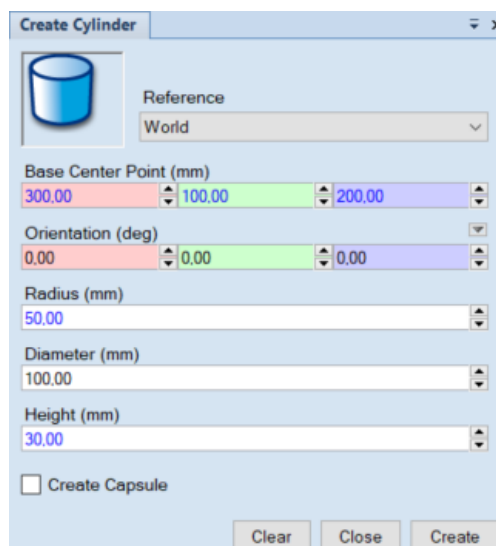
V horní části okna programu, jak lze vidět na Obr.13, se nachází sada sedmi hlavních ovládacích palet, tzv. ribbon. Pod záložkou *File* najdeme základní správu otevřeného řešení, můžeme zde vytvořit nový projekt nebo se připojit k reálnému řídicímu systému, pokud se nachází v naší síti. Hlavní paleta *Home* obsahuje přístup do knihoven s roboty a objekty, možnost vytváření vlastních objektů, definování pozic v prostoru, tzv. targetů, tvorbu cest a základní správu vytvořeného modelu. V paletě *Modeling* se nachází nástroje pro tvorbu a správu objektů v simulaci. Pod následující záložkou *Simulation* najdeme možnosti samotné simulace, záznamu probíhající simulace a analýzu signálů uvnitř řídicí jednotky. Další palety *Controller* a *Rapid* se týkají přímo řídicí jednotky, správy signálů a obsahují nástroje usnadňující tvorbu a optimalizaci

samotného řídicího programu. Poslední záložka *Add-ins* obsahuje možnosti importu a instalace doplňků a modulů do programu RobotStudio online.



Obr. 13: Prostředí programu RobotStudio s vloženým modelem robota IRB 120

2.3.1 Objekty



Obr. 14: Ukázka dialogového okna při tvorbě objektu tvaru válce.

Přímo v prostředí RobotStudia lze vytvářet jednoduché objekty základních geometrických tvarů a přiřazovat jim fyzikální vlastnosti k upřesnění simulace, jako je

hmotnost a osově momenty setrvačnosti, nebo lze využít funkci CAD Import k vložení objektů v běžných CAD formátech jako je IGES, STEP či SAT. Poslední možností je použití online knihoven, které obsahují vybavení od ABB, kterým lze doplnit robotické pracoviště, jako jsou zábrany, pracovní stoly k montáži robotů apod.

2.3.2 Nástroj

Jedním z vložených objektů je nástroj robota. Ten může být, jako kterýkoli jiný objekt, buď importován z knihovny, vytvořen přímo v RobotStudiu nebo vložen ve formě CAD modelu z počítače. K vytvoření nového nástroje s užitím vlastní geometrie slouží funkce CreateTool. Pomocí této funkce si uživatel v dialogovém okně nastaví jméno nového nástroje, z kterého objektu bude nástroj vytvořen a zadá jeho fyzikální vlastnosti, jako je hmotnost, těžiště a jednotlivé osově kvadratické momenty. V dalším kroku se nastaví ToolCenterPoint, neboli pozice koncového bodu nástroje, který se bude pohybovat po cestách a dosahovat jednotlivých targetů.

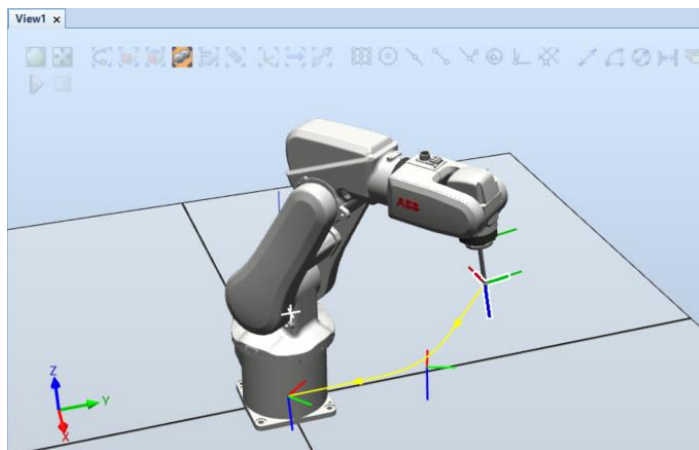
Po vytvoření nového nástroje je možné jej nahrát do knihovny kde bude k dispozici pro použití v jiném projektu. K poslednímu kroku, umístění nástroje na koncovou přírubu robota, jej stačí pomocí funkce drag&drop „přetáhnout“ v levém pracovním okně označeném jako Layout na zde uvedeného robota, případně využít funkci AttachTool k provedení téhož pomocí dialogového okna.

Obr. 15: Úvodní dialogové okno při tvorbě nového nástroje

2.3.3 Cesty, pohyby a targety

Pohyby robota se definují pomocí cest. Cesta se skládá z pohybů mezi targety. Pohyb může být lineární, značený jako MoveL, obecný, značený jako MoveJ, kdy robot sám volí nejrychlejší cestu a kruhový, značený jako MoveC. Ke každému z jednotlivých pohybů lze nastavit rychlost a tzv. zónu, což je požadovaná minimální přesnost v jednotlivých koncových bodech pohybu. Rychlost 150 mm/s se značí jako v150 a zóna o rozměru 30 mm jako z30. Cestu lze zadat buď manuálně pomocí přesného definování souřadnic jednotlivých targetů, nebo lze využít funkci AutoPath, kdy je cesta vytvořena

automaticky, podle hrany objektu nebo jiné křivky vložené do prostředí programu. Důležitou součástí tvorby cest je správná volba orientace nástroje v jednotlivých bodech cesty. Orientaci lze zadat ručně nebo využít nástroje AutoConfiguration, kdy software určí orientaci automaticky a uživatel pouze zkontroluje její správnost.

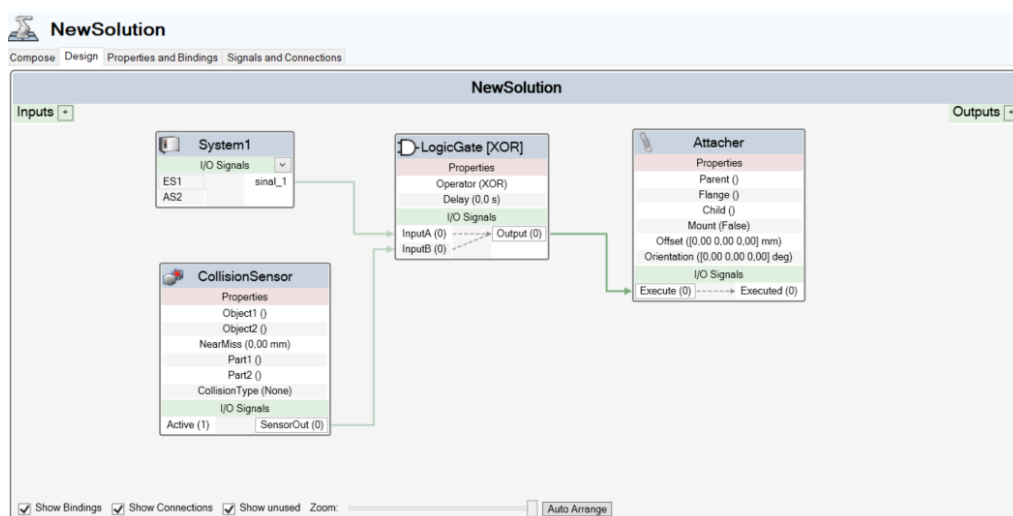


Obr. 16: Cesta tvořená dvěma pohyby mezi třemi targety

Součástí cest nemusí být pouze pohyby, ale mohou obsahovat také tzv. ActionInstructions, neboli události které mají mezi jednotlivými pohyby nastat. Takovou událostí může být například setrvání určitý čas na místě, nastavení výstupního digitálního či analogového signálu, nebo naopak setrvání na pozici do doby, než určený signál nabyde požadované hodnoty.

2.3.4 I/O Signály

Komunikaci mezi řídicím systémem robota, nástrojem a FlexPendantem zajišťují digitální nebo analogové signály označované dle typu jako Input či Output.



Obr. 17: Ukázka grafického znázornění komponent propojených signály

Nový systém obsahuje po prvním spuštění 42 základních signálů, které uživatel nemůže měnit. Jedná se například o signály nouzového zastavení, nebo signály ovládání brzd a motorů jednotlivých členů robota. Uživatel může dle potřeby definovat nové signály a s jejich pomocí ovládat například nástroj připojený na robota, nebo samotnou simulaci v programu RobotStudio. Pomocí modulu nazvaného Station Logic je možné vytvářet komponenty jako například senzory kolize, časovače, či komponenty zajišťující spojení dvou objektů a tím simulující například uchopení výrobku nástrojem robota. Tyto komponenty lze vzájemně propojit pomocí signálů, a tím například zajistit, že pokud senzor zaznamená kolizi s podložkou, dojde k simulaci položení uchopeného objektu, apod. K práci se signály a jejich propojování je možné použít přehledné grafické rozhraní. Některé komponenty mají charakter logického výrazu a umožňují tak se signály vykonávat operace Booleovy algebry.

2.3.5 Rapid

Poté, co uživatel vytvoří pro robota cesty a pohyby, je nutné je nahrát do samotné řídicí jednotky. K tomu slouží funkce Synchronize to RAPID, která vytvořenou souslednost převede na samotný řídicí program. Ten je zapsán pomocí programovacího jazyka RAPID, založeného na programovacím jazyce C#. Program RobotStudio obsahuje přehledný editor RAPIDu zvýrazňující syntax, obsahující funkce jako je Find/Replace nebo základní nástroje pro ladění hotového programu a debugging. V případě, že se uživatel rozhodne tvořit řídicí program přímo v RAPIDu, což se u složitějších úloh stává nutností, je po provedení úprav nutné spustit inverzní funkci Synchronize to Station, která cesty, targety a události vytvořené v RAPIDu vytvoří také v klasickém grafickém prostředí programu.

```
IF pillar = 1 THEN
    MoveL pn2,v150,z100,tool_gripEndAbb;
    MoveL Offs(p2, 0, 0, 30*(n-y)),v150,fine,tool_gripEndAbb;
    by_attach;
    MoveL pn2,v150,z100,tool_gripEndAbb;
    MoveL pn1,v150,z100,tool_gripEndAbb;
    MoveL Offs(p1, 0, 0, 30*(n-x)),v150,fine,tool_gripEndAbb;
    by_detach;
    MoveL pn1,v150,z100,tool_gripEndAbb;
ELSE
    MoveL pn3,v150,z100,tool_gripEndAbb;
    MoveL Offs(p3, 0, 0, 30*(n-y)),v150,fine,tool_gripEndAbb;
    by_attach;
    MoveL pn3,v150,z100,tool_gripEndAbb;
    MoveL pn1,v150,z100,tool_gripEndAbb;
    MoveL Offs(p1, 0, 0, 30*(n-x)),v150,fine,tool_gripEndAbb;
    by_detach;
    MoveL pn1,v150,z100,tool_gripEndAbb;
ENDIF

WaitUntil \Inpos, pause_signal = 0;
ukonceni;
```

Obr. 18: Ukázka syntaxe RAPIDu

RAPID pracuje se základními proměnnými jako je *num*, *string* nebo *bool* a přidává také další, takzvaně komplexní datové typy. Jedním z nich je proměnná typu *robtarget*, určující nejen přesný bod v prostotu, ale také orientaci nástroje robota a natočení jednotlivých os robota ve chvíli jeho dosažení. Dalšími komplexními datovými typy jsou například proměnné typu *pos* určující pouze pozici v prostoru nebo *orient*, specifikující natočení nástroje.

Celý řídicí program se skládá z do sebe vnořených procedur, přičemž hlavní a počáteční procedurou je procedura označená jako Main. RAPID podporuje základní podmínky a cykly, jako je IF, FOR, WHILE a další. V programu se také mohou vyskytovat speciální příkazy, například pro přiřazení hodnoty některému signálu, nebo naopak příkaz pozastavující chod programu do doby, než zvolený signál nabyde předem definované hodnoty. [30]

2.3.6 Simulace

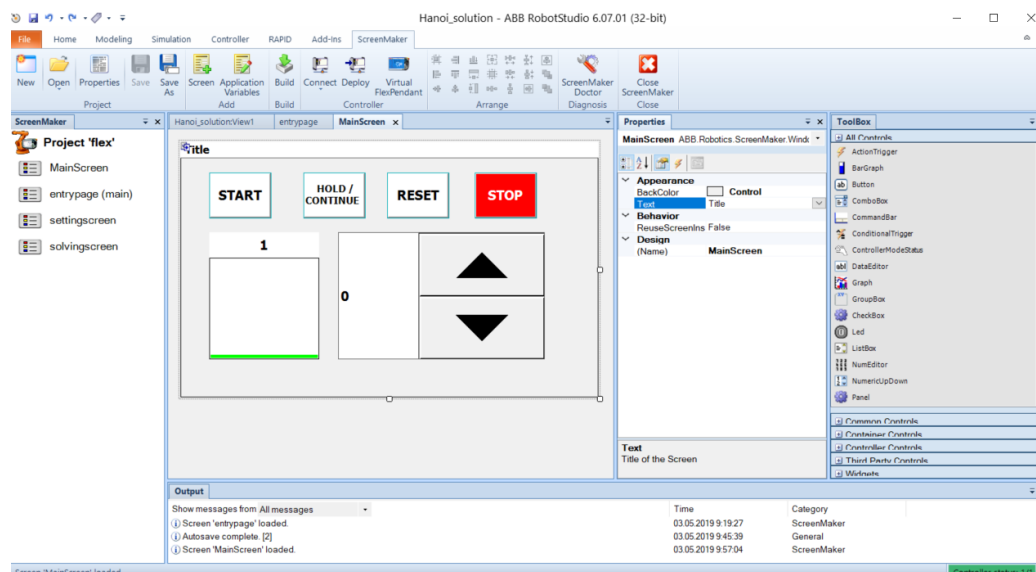
Po vytvoření celé stanice, řídicího programu, jednotlivých signálů a funkčních komponent nabízí RobotStudio možnost celou úlohu simulovat přesně tak, jak se bude řídicí program spolu se signály chovat na reálném hardware. Kromě kvalitní vizualizace úlohy nabízí také pokročilé funkce analýzy jednotlivých signálů, natočení os, spotřeby elektrické energie a dalších parametrů. Zaznamenaná data lze jednoduše interpretovat pomocí grafů nebo exportovat do tabulkového procesoru k důkladnější analýze. K přehledné kontrole hodnot jednotlivých proměnných během simulace slouží funkce RapidWatch. Umožňuje také zvolit dvojici objektů, a v průběhu simulace analyzovat jejich vzdálenost pomocí funkce Create Collision Set. Celou simulaci lze také nahrávat do souboru mp4, v případě potřeby zaznamenávat i celé prostředí aplikace v průběhu simulace.

2.3.7 ScreenMaker

ScreenMaker je nástroj v RobotStudiu určený k vývoji vlastních aplikací s GUI, neboli grafickým uživatelským rozhraním pro FlexPendant, umožňující tvorbu těchto aplikací bez znalosti vývojového prostředí Visual Studio a programování na platformě .NET.

GUI uživateli výrazně ulehčuje práci s průmyslovými roboty, nahrazením vnitřních mechanismů řídicí jednotky robota jejich grafickou reprezentací. GUI se skládá z několika oken, které se spouští v ovládacím prostoru na dotykové obrazovce FlexPendantu. Typické ovládací okno se skládá z několika menších grafických komponentů jako jsou tlačítka, obrázky a textová pole, pomocí nichž uživatel s aplikací pracuje. Interakce s rozhraním, jako je například stisknutí tlačítka, se nazývá událost. Každý komponent vyvolává specifickou událost, která je předána aplikaci ke zpracování. GUI aplikace reaguje na jednotlivé události v pořadí, v jakém je uživatel zadá. Běh samotné aplikace je tedy více ovlivňován jednotlivými událostmi, než že by byl sekvenční od začátku do konce. Protože nelze předvídat úkony uživatele pracující s aplikací, musí

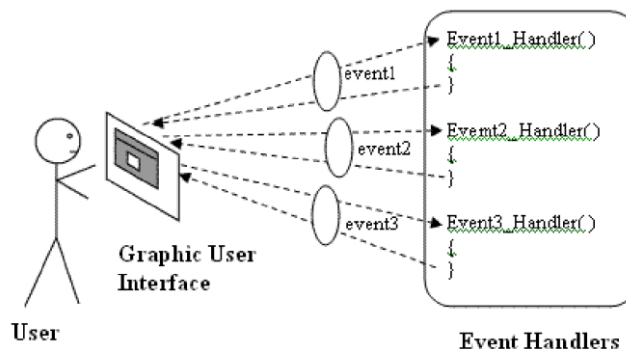
aplikace správně fungovat bez ohledu na to, v jakém pořadí či množství uživatel události vyvolá. Události, které nejsou v daný moment relevantní, musí aplikace ignorovat.



Obr. 19: Ukázka pracovního prostředí nástroje ScreenMaker, uprostřed v pracovním okně je znázorněn náhled jednoho z budoucích oken aplikace.

Aplikace obsahuje instrukce, přiřazené k jednotlivým událostem. Takovou instrukcí může být sepnutí I/O signálu, zápis či čtení proměnné v RAPIDu, zpracování logických operací, nebo jiné výpočty. Z pohledu vývojáře se tedy GUI aplikace skládá ze dvou celků, zobrazovací části, kterou vidí uživatel a která vyvolává jednotlivé události a procesní části, která jednotlivé události zpracovává.

Je třeba si uvědomit, že samotná řídicí jednotka robota a ovládací FlexPendant jsou dva různé počítače, běžící na dvou různých operačních systémech. Aplikace, vytvořené nástrojem ScreenMaker pro FlexPendant, mají tedy vlastní systém proměnných a jsou nezávisle přenositelné mezi více zařízeními. Samotný FlexPendant může obsahovat větší množství takových aplikací, které si uživatel sám spouští pomocí menu. Nástroj ScreenMaker umožňuje nahrání hotové aplikace do připojeného, ať už reálného, nebo pouze virtuálního FlexPendantu. Pomocí virtuálního FlexPendantu lze vytvořené aplikace otestovat před jejich reálným nasazením. [31]



Obr. 20: Schéma struktury GUI aplikace pro FlexPendant [31]

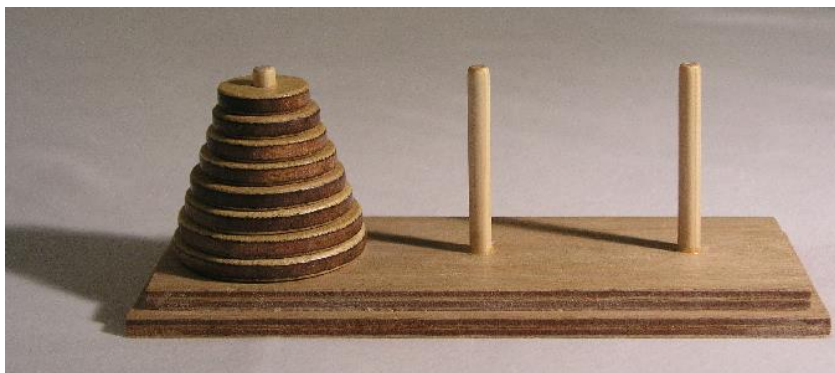
3 VLASTNÍ ŘEŠENÍ ÚLOHY

3.1 Výběr laboratorní úlohy

Po dohodě s vedoucím bakalářské práce bylo rozhodnuto, že řešenou laboratorní úlohou bude logický hlavolam hanojské věže, jakožto typický příklad jednoduché manipulační úlohy vyžadující k vyřešení velké množství cyklů, a tedy úlohy vhodné k automatickému řešení.

3.1.1 Hanojské věže

Hanojské věže jsou matematicko-logický hlavolam, který vymyslel francouzský matematik Édouard Lucas a poprvé jej publikoval v roce 1883. Skládá se ze tří kolíků, věží, na které lze navlékat ploché disky o různých průměrech seřazené od největšího (dole) po nejmenší (nahore). Na začátku úlohy jsou všechny disky naskládány na prvním kolíku a úkolem řešitele je přemístit postupně všechny kotouče na jeden ze dvou zbývajících kolíků. Poslední kolík slouží k dočasnému odkládání disků. V průběhu řešení je nutno dodržovat tři základní pravidla: V průběhu jednoho tahu lze přemístit pouze jeden disk, přemísťuje se vždy disk z vrcholu věže na vrchol jiné věže a je zakázáno položit disk s větším průměrem na disk s průměrem menším. Hlavolam se také vyskytuje ve verzi bez kolíků, přičemž jednotlivé disky se na sebe volně pokládají a tvoří tři sloupce. [32]



Obr. 21: Dřevěný hlavolam hanojské věže s osmi disky [33]

3.1.2 Možnosti řešení

Tento hlavolam lze řešit několika základními způsoby. První způsob předpokládá střídavé přemísťování nejmenšího disku a jiného disku než nejmenšího. Přesune-li se nejmenší disk, pak se vždy přesune o jednu věž dál ve stejném směru, a to doprava při celkovém sudém počtu disků a doleva při lichém (předpokládáme že věže stojí v řadě vedle sebe a počáteční je nejvíce vlevo a cílová nejvíce vpravo). Pokud se nejmenší disk nachází na poslední věži ve směru svého pohybu, přesune se na věž na protějším konci. Má-li být přesunut jiný než disk nejmenší, lze jej vždy přesunout pouze jedním způsobem.

Další způsob řešení se označuje jako řešení pomocí rekurze, a vychází z úvahy, že každé řešení obsahuje právě jeden tah, během kterého se přemísťuje největší disk z počáteční na cílovou věž. K tomu může dojít pouze tehdy, pokud jsou všechny ostatní disky naskládány na třetí věž, označené jako odkládací. Označíme-li počet disků n , tak nejdříve přesuneme $n-1$ disků (všechny kromě největšího) na třetí, odkládací, věž, pak přesuneme největší disk z počáteční věže na cílovou, a nakonec přesuneme opět $n-1$ disků z věže odkládací na věž cílovou. Přesun $n-1$ disků ovšem můžeme vykonat pomocí rekurzivního volání tohoto algoritmu pro $n-1$ disků, protože na největší disk můžeme položit kterýkoli jiný a v řešení úlohy nám nijak nepřekáží. Z tohoto řešení lze také matematickou indukcí dokázat, že nejmenší možný počet tahů potřebný k vyřešení úlohy s n disky je $2^n - 1$ tahů. [32, 34]

Třetí nastíněný způsob řešení se označuje jako řešení iterativní, a je jistou modifikací prvně popsaneho způsobu. Označíme-li věže jako A – počáteční, B – odkládací a C – cílová, tak při dodržování základních pravidel úlohu vyřešíme opakováním následujících tří kroků:

Pro sudý počet disků:

- Proved'te přesun disku mezi věžemi A a B
- Proved'te přesun disku mezi věžemi A a C
- Proved'te přesun disku mezi věžemi B a C

Pro lichý počet disků:

- Proved'te přesun disku mezi věžemi A a C
- Proved'te přesun disku mezi věžemi A a B
- Proved'te přesun disku mezi věžemi B a C

Přesunem disku se myslí jeho pohyb v obou směrech mezi výše uvedenými věžemi. Při dodržování tohoto postupu až do umístění všech disků na cílové věži dojde k vyřešení úlohy v nejmenším možném počtu tahů. [34]

3.1.3 Volba použitého řešení

Po rozvaze vhodnosti jednotlivých řešení k implementaci do řídicího programu robota bylo rozhodnuto, že bude použit třetí uvedený způsob, který byl vyhodnocen jako nejjednodušší a díky své sekvenčnosti nejsnáze implementovatelný do programovacího jazyka RAPID.

3.2 Řešení úlohy

V průběhu řešení úlohy postupně docházelo k definování jejich přesných parametrů. Výsledné řešení lze popsat následovně: Robot řeší úlohu hanojské věže na základě základních vstupních dat, které uživatel zadá pomocí ovládací aplikace spuštěné na FlexPendantu. Vstupními parametry jsou počet disků a informace, na kterou věž má dojít k přeskládání disků. Po zadání vstupních parametrů úlohy a jejím spuštění lze pomocí ovládací aplikace řešení pozastavit nebo úplně zrušit. Po dokončení úlohy, nebo jejím zrušení v průběhu řešení, je systém připravený řešení zopakovat, a to se stejnými vstupními parametry, jako v průběhu předchozího řešení, pokud mu uživatel před samotným spuštěním nezadá parametry nové. Jednotlivé disky robot uchopuje ve středu jejich horní strany, pomocí podtlakové přísavky efektoru upevněného na koncové přírubě.

Ačkoli samotný ovládací program je univerzální a úlohu dokáže řešit pro libovolný počet disků, teoreticky omezený pouze limity jednotlivých datových typů použitých pro proměnné, pro potřeby simulace byla volba tohoto vstupního parametru omezena na 0-5 disků. Bylo tak učiněno z toho důvodu, že jednotlivé objekty (disky) a k nim se vztahující ovládací signály musí být dopředu vytvořeny a připraveny v simulačním prostředí. Jejich automatickou tvorbu až na základě vstupních parametrů od uživatele program nepodporuje. V případě, že by navržený řídicí program nepočítal s použitím v komplexní simulaci, ale byl určený pouze k použití na reálném hardware, nebylo by toto omezení počtu zadaných disků nutné. Úprava výsledného programu do stavu k použití pouze na reálném hardware bez limitu počtu řešených disků by byla spíše otázkou odstranění některých jeho částí a celkového zjednodušení než dalšího rozvoje.

3.2.1 Návrh prostředí úlohy

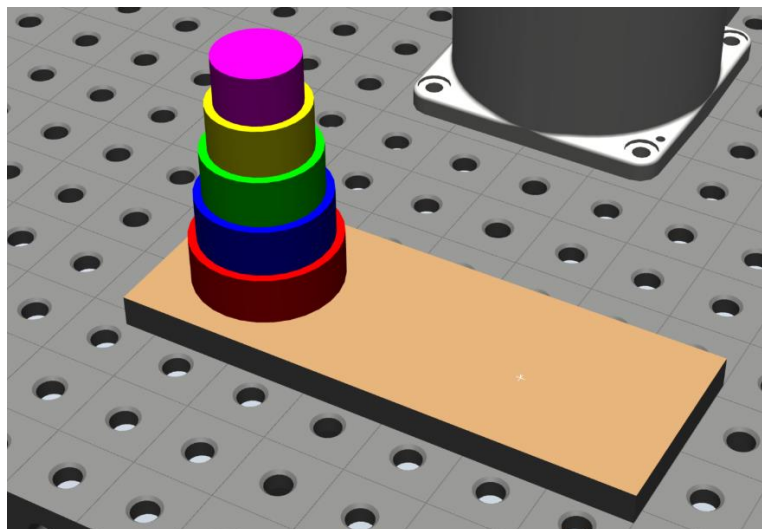
Podoba simulačního prostředí byla zvolena jako jednoduchá a názorná. Robot ABB IRB 120 je osazen ve středu profesionálního pracovního stolu Siegmund o půdorysu 1500x1000 mm s pracovní plochou ve výšce 830 mm nad úroveň podlahy, stejně jako jeho reálný protějšek v laboratoři. Komponenty hlavolamu jsou podstava o rozměru 320x120 mm a 5 disků, jejichž rozměry jsou uvedeny v tab. 3.

Tab. 3: Rozměry jednotlivých disků v úloze

Označení	Disk 1	Disk 2	Disk 3	Disk 4	Disk 5
Průměr	90 mm	80 mm	70 mm	60 mm	50 mm
Výška	30 mm				

Jednotlivé disky jsou na počátku úlohy i v jejím průběhu umístovány soustředně, na tři osy odpovídající třem věžím od sebe vzdálené 100 mm. Celý hlavolam je položený podélně na pracovní desce před robotem ve vzdálenosti 150 mm a při pohledu na robota zepředu je posunut o 40 mm vpravo, aby se nenacházel přesně v ose robota, což by mohlo vést k problémům se singularitou. Jako výchozí věž, na které jsou disky umístěny před začátkem úlohy, byla zvolena věž jevící se jako na levé straně podstavy. Pohled na model

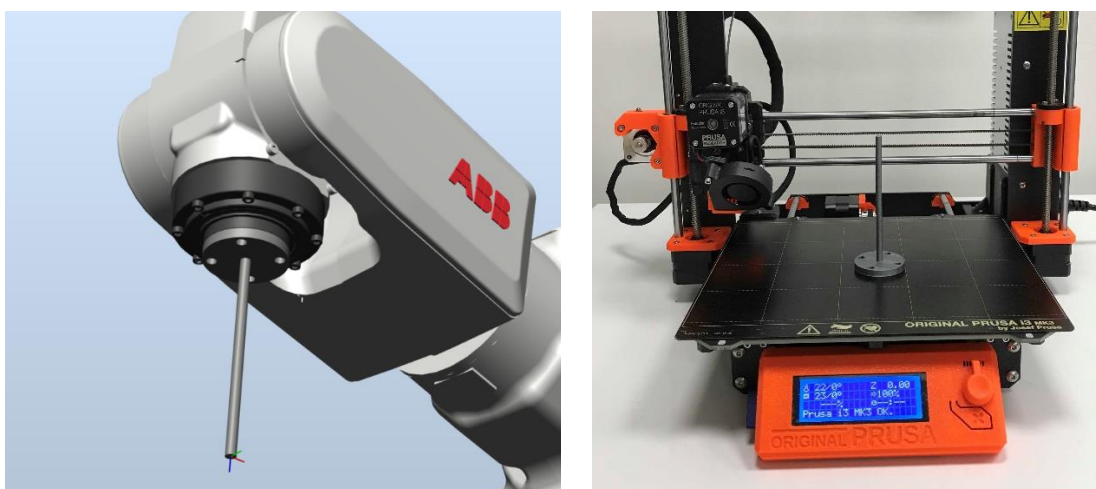
hlavolamu se nachází na obr. 21. Celé prostředí si lze prohlédnout v interaktivním programu v příloze.



Obr. 22: Pohled na model hlavolamu hanojské věže ve výchozí pozici ve verzi bez středících kolíků

3.2.2 Návrh efektoru

Efektor byl navržen tak, aby svými celkovými rozměry odpovídal případnému reálnému efektoru a zároveň byl pro potřeby testu na reálném hardware jednoduše vytisknutelný na 3D tiskárně. K připevnění na koncovou přírubu robota pomocí šroubů byl opatřen čtyřmi dírami. Po vytvoření modelu efektoru pomocí 3D CAD programu byl importován do RobotStudia a byl z něj standartním postupem vytvořen nástroj pomocí funkce CreateTool, který byl posléze osazen na robota.



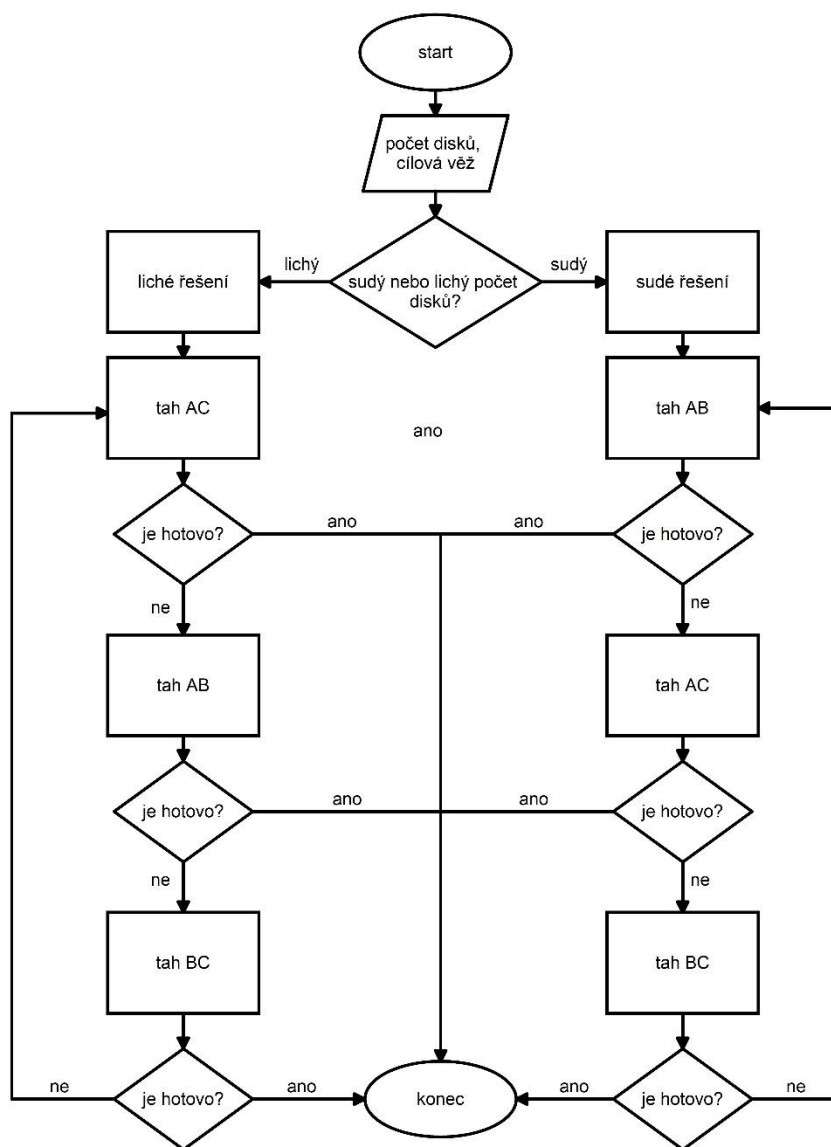
Obr. 23: Model efektoru osazený na koncové přírubě robota (vlevo) a skutečný efektor po vytištění na 3D tiskárně (vpravo)

3.2.3 Návrh řídicího programu

S ohledem na zajištění funkčnosti celé úlohy a simulace musí řídicí program obsahovat především následující funkcionality:

- Řešení hlavolamu vybraným způsobem
- Čtení a záznam aktuálního stavu úlohy
- Řízení prvků simulace
- Komunikace s ovládací aplikací na FlexPendantu

Vybraný algoritmus řešení byl již popsán na předešlých stranách, jeho přenesení do formy programu je zobrazeno na vývojovém diagramu níže.

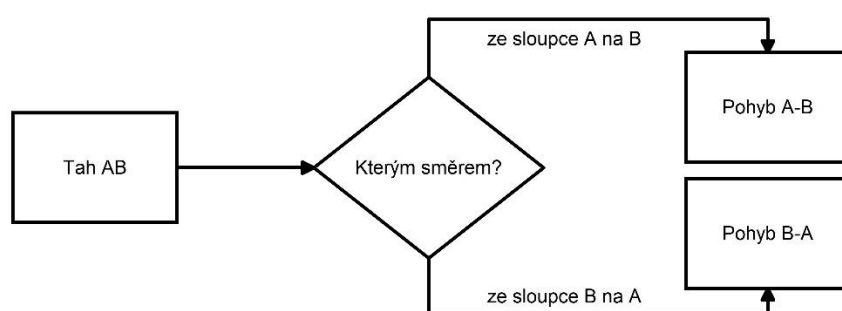


Obr. 24: Zjednodušený vývojový diagram řídicího programu

RAPID lze jednoduše členit do jednotlivých procedur a ty do sebe vzájemně vnořovat. V našem případě nejprve program zjistí, zda je počet řešených disků lichý nebo sudý, a v závislosti na tom spustí proceduru pro liché, nebo sudé řešení. Tyto hlavní

procedury obsahují každá tři základní tahy s tím, že po každém vykonaném tahu dochází ke kontrole, zda není již celá úloha vyřešena. Díky tomu, že počet kroků nutných k vyřešení úlohy lze spočítat z počtu disků, probíhá kontrola jednoduchým odčítáním počtu zbývajících kroků.

Jednotlivé základní tahy jsou procedury vnořené do procedur sudého nebo lichého řešení. Jelikož přesun probíhá mezi třemi věžemi tak základní tahy jsou také tři, a to tah AB, tah AC a tah BC. Z podstaty zvoleného algoritmu řešení hanojských věží může například tah AB znamenat přemístění z věže A na věž B i naopak, podstatné je v takovou chvíli to, že lze provést pohyb pouze jedním z těchto dvou směrů. K tomu, aby program zjistil, kterým směrem bude aktuální tah probíhat, potřebuje zajišťovat svou druhou základní funkci, a to postihovat a rozpoznávat aktuální stav úlohy.



Obr. 25: Diagram procedury tahu AB

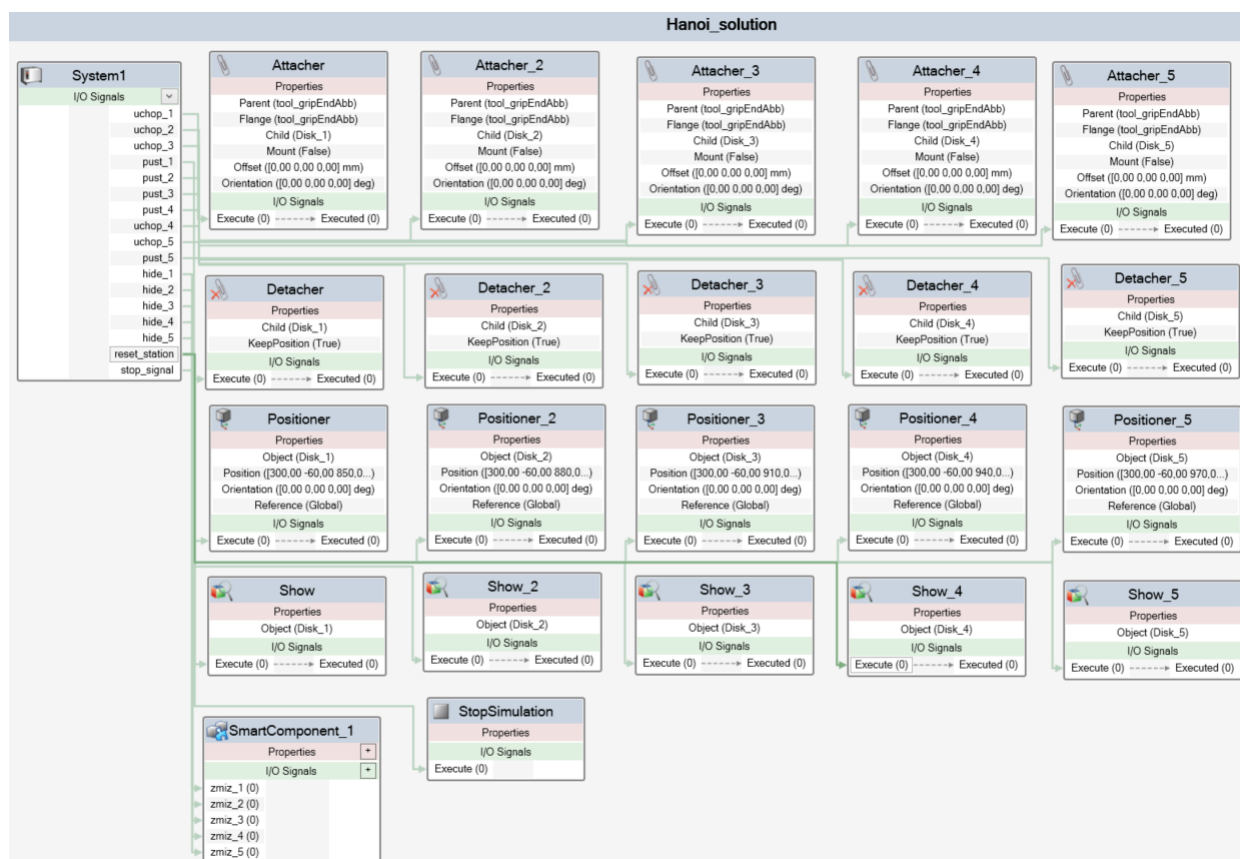
K účelu zjišťování stavu úlohy jsou při spuštění programu vytvořeny tři stavové řádkové matice, každá pro jednu z věží. Jednotlivé prvky matic odpovídají jednotlivým možným umístěním disků v prostoru. V případě, že se na dané pozici nenachází žádný disk, má daný prvek hodnotu $n+1$ za předpokladu, že n je celkový počet disků. V případě že se na pozici nachází disk, nabývá prvek hodnoty celých kladných čísel, přičemž hodnotu můžeme popsat jako kolikátý nejmenší disk v úloze se na dané pozici nachází. Nejmenší disk tedy postihuje hodnota 1, největší n , druhý nejmenší 2 atd. Jednotlivé procedury základních tahů AB, AC a BC obsahují soubor cyklů, během nichž dochází k procházení stavových matic. Nejdříve se zjistí, na které pozici se nachází nejvýše položené disky na věžích, mezi kterými má proběhnout výměna, a potom program porovná hodnoty stavových matic pro konkrétní pozice a z toho určí kterým směrem se bude přesun disku konat. Následně vyvolá proceduru pro konkrétní pohyb, tedy například z věže A na věž B, a nakonec provede úpravy stavových matic tak, aby odpovídaly aktuálnímu stavu.

Tab. 4: Hodnoty stavových matic při spuštění úlohy s pěti disky

Název stavové matice	Hodnota matice
a	1,2,3,4,5
b	6,6,6,6,6
c	6,6,6,6,6

Pro vykonání pohybu mezi věžemi je zavedeno 6 procedur, postihujících všechny možné případy, které mohou nastat – z věže A na věž B, z věže B na věž A, z věže A na věž C apod. Tyto procedury obsahují konkrétní pohybové instrukce pro robota, obsahující mimo jiné také rychlost pohybu a zónu. Jedna z těchto procedur byla zobrazena jako ukázka syntaxe RAPIDu v podkapitole 2.3.5 na Obr.17. Kromě samotného pohybu robota je také potřeba zajistit ovládání efektoru, v případě simulace se tedy dostáváme k třetí důležité funkcionalitě řídicího programu, a to k řízení prvků simulace.

Komunikace mezi řídicím programem a simulačním prostředím probíhá pomocí digitálních I/O signálů zavedených v řídicím systému robota. Jednotlivé signály jsou propojeny s jednotlivými komponenty simulace, které zajišťují jednotlivé funkcionality, jako je uchopení a položení disku, či skrytí disků, které v zadané úloze nejsou potřeba. Propojení signálů a komponent lze vidět na obr. 25 zobrazujícím StationLogic řídicího programu. Pro každý disk je potřeba vytvořit komponentu *Attacher*, zajišťující jeho uchopení, *Detacher* zajišťující položení, *Positioner* zajišťující navrácení disků na výchozí pozice po dokončení úlohy a *Show* určující, zda je disk při úloze viditelný, či nikoli.



Obr. 26: StationLogic navrženého programu

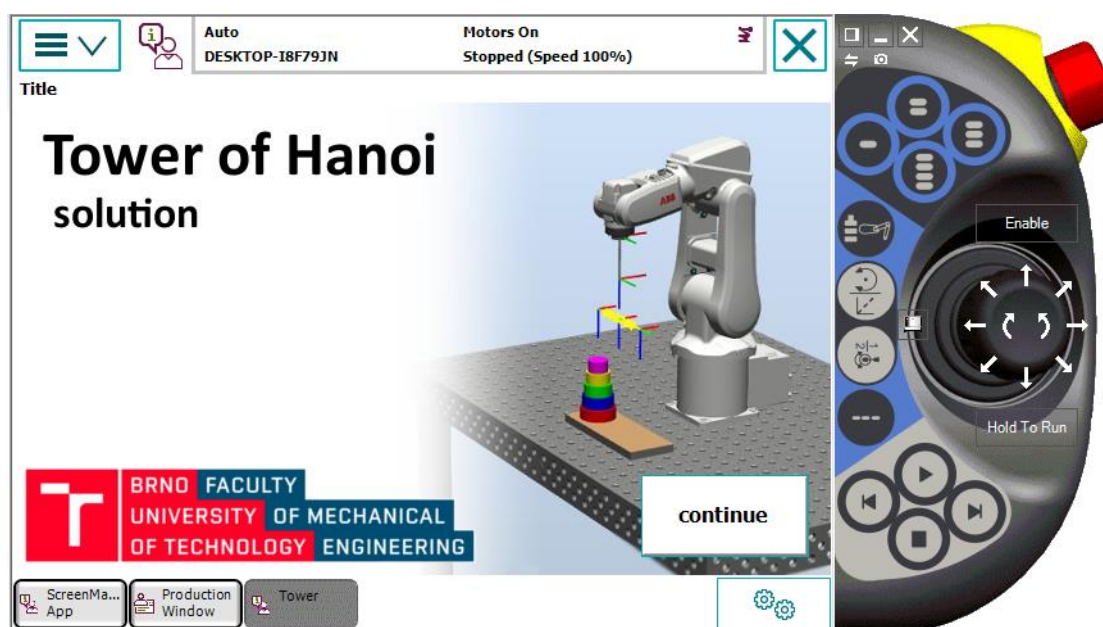
I/O signály přímo souvisí také se čtvrtou důležitou funkcionalitou řídicího programu, kterou je komunikace s ovládací aplikací, která je spuštěná na FlexPendantu. Tato komunikace probíhá jednak ve formě I/O signálů, například na začátku úlohy program čeká na signál zahajující její řešení, anebo naopak kontroluje po každém tahu,

zda nebyl vydán signál k jejímu přerušení. Druhým způsobem komunikace je zápis a čtení proměnných v RAPIDu. K tomu dochází například ve chvíli, kdy uživatel zadá počet disků (hodnota proměnné n), nebo když volí, na kterou věž mají být disky přeskládány.

Celkem program pracuje s 22 proměnnými, obsahuje 24 procedur a ke komunikaci se simulačním prostředím a FlexPendantem využívá 19 digitálních signálů.

3.2.4 Tvorba uživatelského rozhraní

Uživatelské rozhraní je jedním z klíčových prvků úlohy, která by v případě jeho nevytvoření musela být ovládána ruční změnou parametrů přímo v RAPIDu, což by podstatným způsobem zkomplikovalo testování a prezentaci úlohy. Díky grafickému uživatelskému rozhraní mohou laboratorní úlohu spouštět, testovat a ovládat také uživatelé, kteří nejsou přímo seznámeni s konstrukcí a principem samotného řídicího programu.



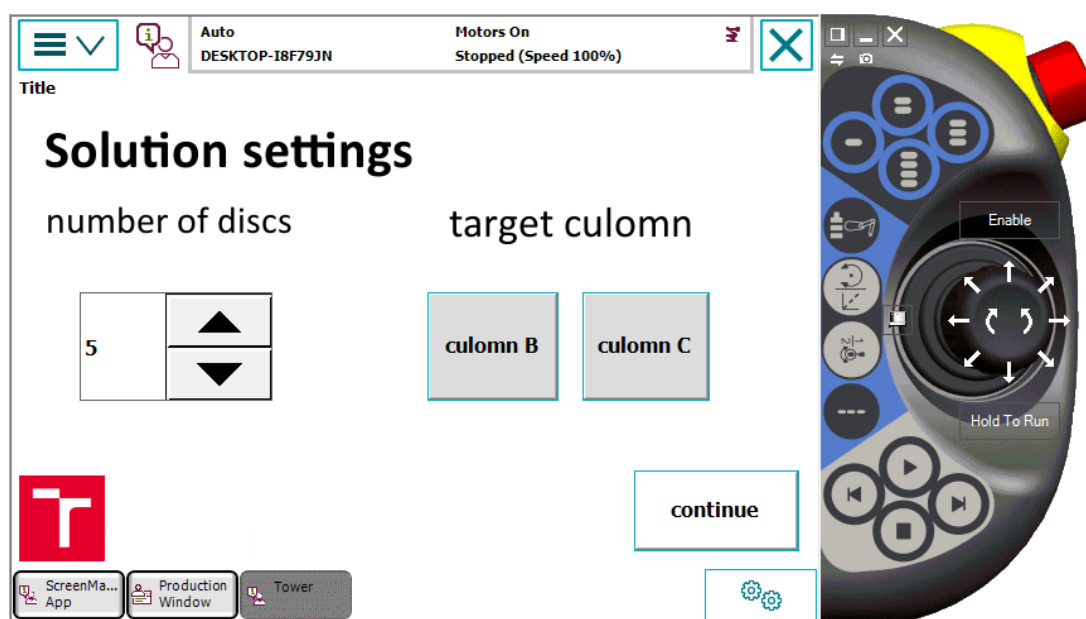
Obr. 27: Úvodní stránka ovládací aplikace spuštěná na virtuálním FlexPendantu

Ke tvorbě grafického uživatelského rozhraní byl použit nástroj ScreenMaker, který je jedním z doplňků programu RobotStudio. Ovládací aplikace byla navržena jako sestava tří ovládacích stránek, mezi kterými se uživatel pohybuje pomocí přepínacích tlačítek. Popis jednotlivých stránek se nachází v Tab.5.

Tab. 5: Stránky ovládací aplikace

Název stránky	popis
Entry screen (main)	Úvodní stránka, obsahující název úlohy, logo školy, vizualizaci řešené úlohy a navigační tlačítko.
Setting screen	Stránka sloužící k nastavení základních parametrů úlohy, jako je počet disků a cílová věž.
Solving screen	Stránka sloužící ke spuštění a ovládání úlohy. Kromě ovládacích tlačítek obsahuje také informaci o počtu tahů, zbývajících do vyřešení úlohy.

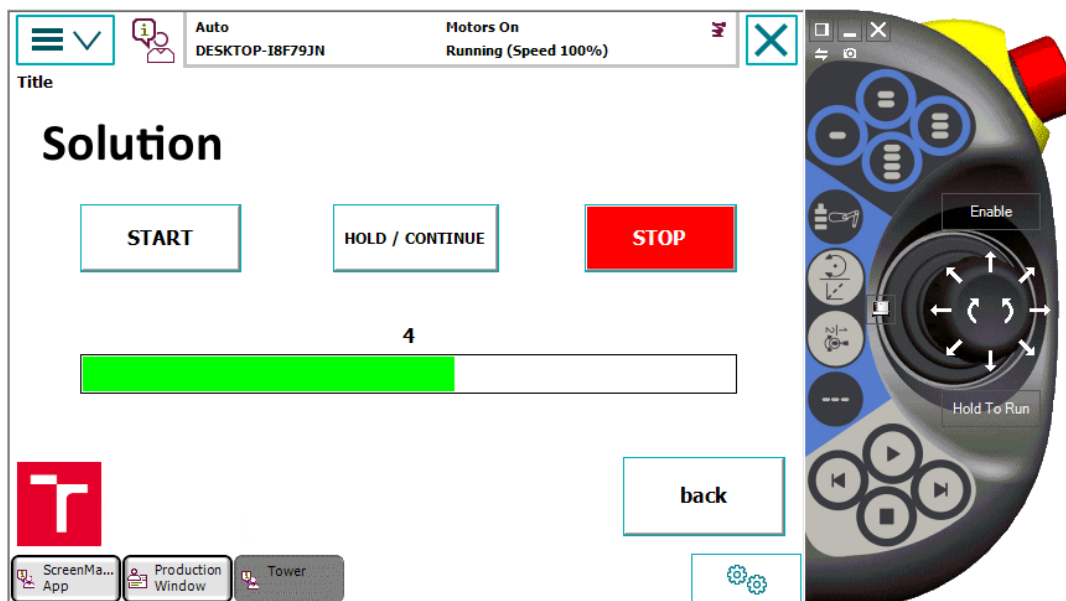
Ovládací aplikace komunikuje s řídicím systémem robota. Ve chvíli kdy uživatel provede změnu nastavení, spustí se událost příslušící tlačítku, které tuto změnu provedlo a tato událost zajistí přepsání hodnoty proměnné v RAPIDu. Tento typ události nastává při použití přepínačů na stránce označené jako Setting screen, která je zobrazená na Obr.27.



Obr. 28: Stránka ovládací aplikace sloužící k nastavení základních parametrů úlohy

Dalšími základními typy událostí jsou čtení proměnné v RAPIDu a nastavení hodnoty I/O digitálního signálu. Tyto typy byly použity při tvorbě stránky ovládací aplikace označené jako Solving screen, zobrazené na Obr. 29. Ovládání běhu úlohy je zajištěno pomocí signálů. Řídicí program čeká na zahájení řešení úlohy do doby, než uživatel pomocí tlačítka start sepne spouštěcí signál. Naproti tomu po jednotlivých tazích kontroluje, zda uživatel nespustil pomocí dalších tlačítek signály pozastavující nebo ukončující řešení úlohy. V případě hrozícího nebezpečí je možné robota zastavit užitím hardwarového tlačítka nouzového zastavení, které se nachází na FlexPendantu. Poslední výrazný prvek stránky Solving screen je vodorovný ukazatel postupu řešení

úlohy, doplněný o číslo ukazující kolik tahů ještě zbývá provést do vyřešení úlohy. Tento prvek načítá dvě proměnné z RAPIDu, a to počáteční a aktuální počet zbývajících kroků.

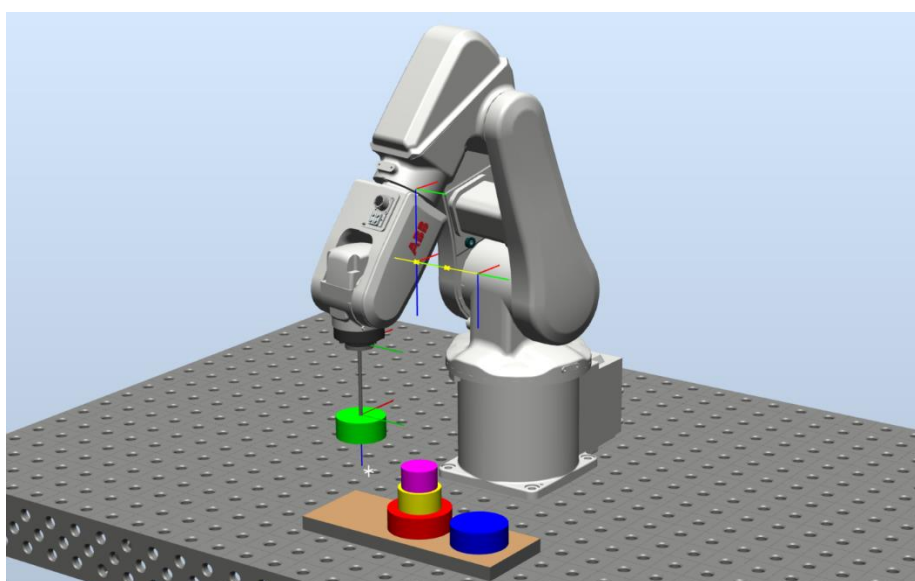


Obr. 29: Stránka ovládací aplikace sloužící k ovládání běhu laboratorní úlohy

4 TESTOVÁNÍ

4.1 Simulace úlohy v programu RobotStudio

Po vytvoření řídicího programu a simulačního prostředí bylo přistoupeno k samotné simulaci a testování programu. Nejdříve proběhla simulace v programu RobotStudio. Za tímto účelem byl v programu spuštěn virtuální řídicí systém, virtuální FlexPendant a samotná testovaná úloha. Nejdříve byly vyzkoušeny základní funkcionality programu a bylo ověřeno, že simulace probíhá dle předpokladů. Ukázka simulačního prostředí v průběhu testů se nachází na Obr. 30.



Obr. 30: Simulační prostředí v průběhu testu úlohy s pěti disky

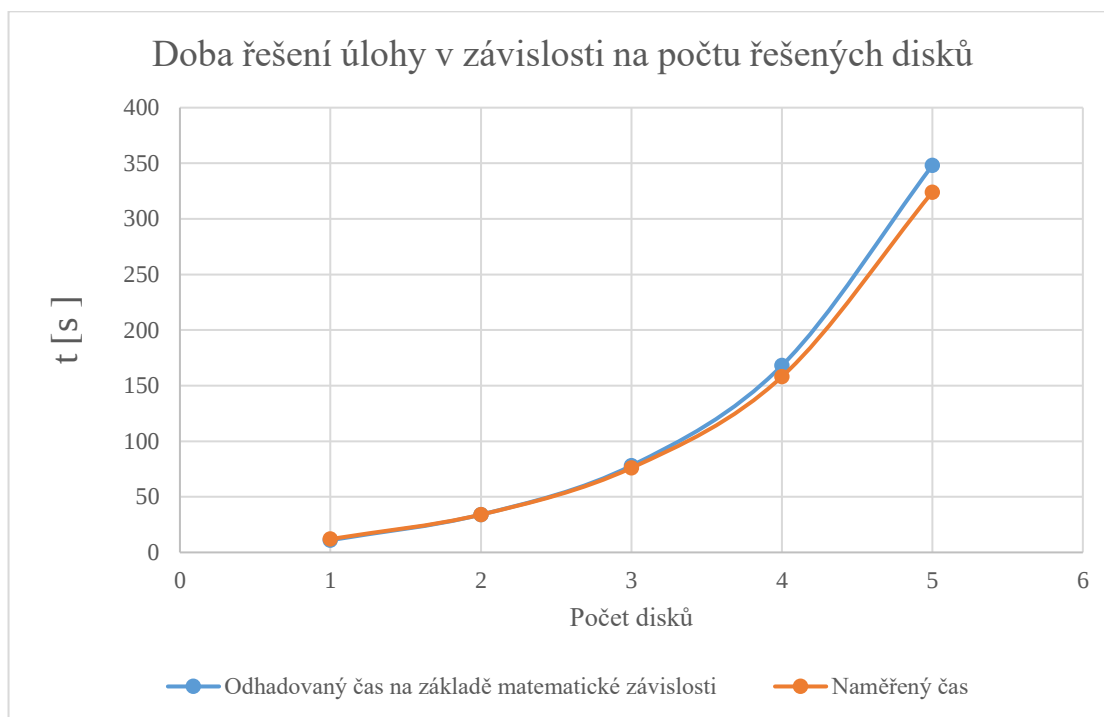
Po ověření základních funkcionalit bylo přistoupeno k měření doby trvání řešení úlohy při různě nastavených vstupních parametrech. Naměřené hodnoty jsou uvedeny v Tab. 6. První sada hodnot odpovídá přesunu na věž B a druhá na věž C.

Tab. 6: Doba řešení úlohy, hodnoty jsou uvedeny v sekundách

	1 disk	2 disky	3 disky	4 disky	5 disků
Přesun na věž B	12	34	76	158	324
Přesun na věž C	13	33	76	158	324

Na naměřených datech si můžeme všimnout několika zajímavých zákonitostí. Při nižším počtu disků se liší čas, za který je úloha vyřešena. To je dáno tím, že od počáteční věže je to na věž C dále, než na věž B. S přibývajícím počtem kroků, nutných k vyřešení úlohy, se tento rozdíl rychle ztrácí. Další zajímavou zjištěnou zákonitostí je, že s přibývajícím počtem řešených disků se snižuje čas, který robot průměrně potřebuje

k vykonání jednoho tahu. To je způsobeno tím, že při řešení většího množství disků jsou jednotlivé věže v průběhu řešení vyšší a robot tak v průběhu tahu urazí kratší vzdálenost. Grafické znázornění tohoto jevu pomocí porovnání naměřené závislosti času řešení na počtu řešených disků s časem vypočteným díky známé závislosti počtu tahů nutných k řešení na počtu disků je zobrazeno na Obr. 31. V případě řešení pěti disků dosahuje tato odchylka více jak 20 sekund.



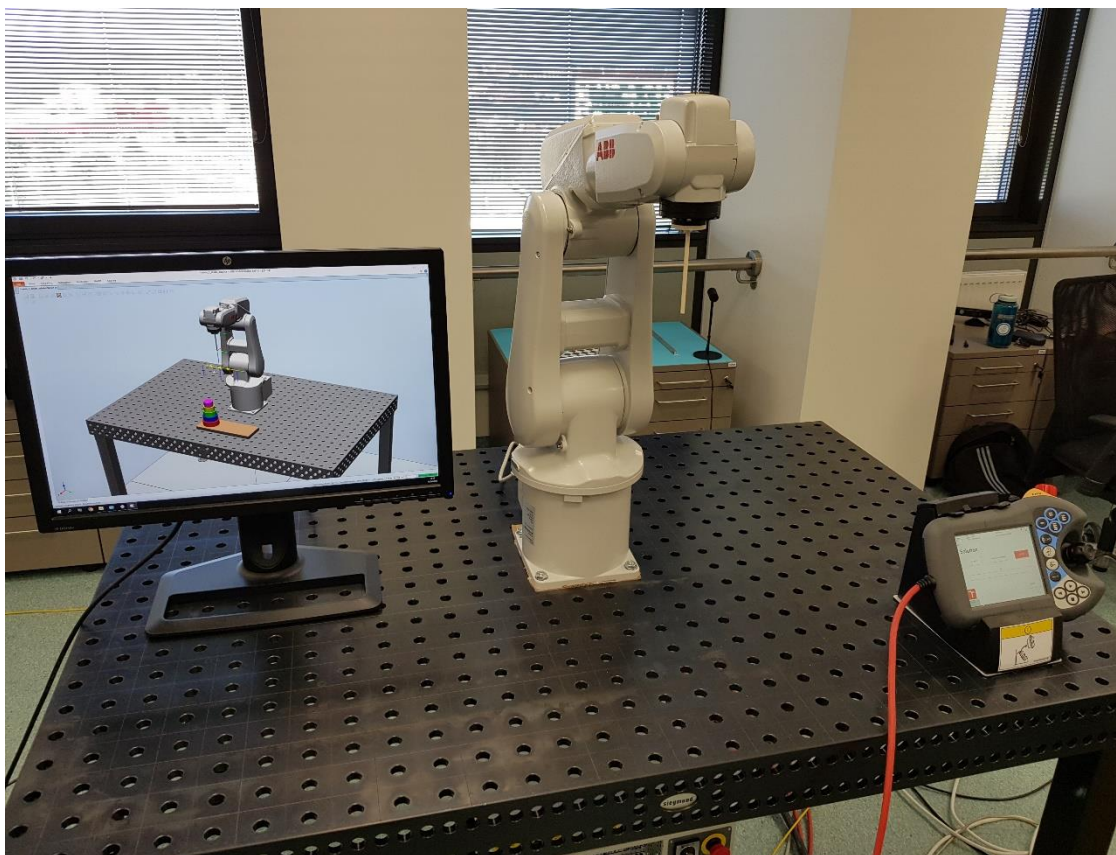
Obr. 31: Doba řešení úlohy v závislosti na počtu řešených disků

Na výše uvedeném obrázku lze také dobře sledovat exponenciální růst času potřebného k řešení úlohy s přibývajícími disky. V případě, že by řešení probíhalo stejnou rychlostí a úloha by obsahovala 10 disků, její dokončení by si již vyžádalo více než 3 hodiny. V případě úlohy s 20 disky již skoro pět měsíců a s 33 více než 1000 let.

V průběhu simulace lze využít nástroj Signal Analyzer, díky kterému lze jednak sledovat průběh a změny jednotlivých signálů, ale také různých jiných zajímavých údajů, jako je natočení jednotlivých os, příkon robota či spotřeba elektrické energie. Jednotlivé grafy s naměřenými hodnotami se nacházejí v příloze A této práce. Špičkový výkon v průběhu řešení se pohybuje okolo 50 wattů a potřebná energie k vyřešení úlohy o třech discích je zhruba 1,6 kJ.

4.2 Ověření úlohy na reálném hardware

Test řídicího programu a ovládací aplikace proběhl ve školní laboratoři pod dohledem vedoucího bakalářské práce. Fotografie testovací stanice v laboratoři se nachází na Obr.32.



Obr. 32: Testovací stanice ve školní laboratoři

Testovací stanice se skládala z pracovního stolu, robota ABB IRB 120, řídicího systému IRC5C, FlexPendantu a sestavy monitoru s počítačem na kterém probíhala paralelně s pohyby robota simulace ve virtuálním prostředí.

Do řídicího systému a FlexPendantu byl nahrán navržený program a přistoupilo se k testům. Video z provedeného testování se nachází na CD v příloze B. V průběhu testů byly vyzkoušeny všechny základní funkcionality programu. Během testování se ukázalo, že vytvořené grafické uživatelské rozhraní je při ovládání robota velkou výhodou a podstatně ulehčuje ovládání řídicího programu. Program fungoval dle předpokladů, čímž se potvrdila přesnost dříve provedené simulace.

5 POUŽITÍ VR TECHNOLOGIÍ

5.1 Technologie virtuální reality

Návrh a vývoj nového stroje, výrobní linky či robotizovaného stanoviště je velmi složitý, a především nákladný proces. V minulosti praktikovaná výroba prototypů, které se dlouze zkouší a testují je drahá a časově náročná. V dnešní době by výsledkem vývoje měl být prototyp, který je již takřka bez závad a splňuje vše očekávané, a právě v tom mohou pomoci technologie virtuální reality.

Virtuální realita je poměrně mladá technologie, jejíž nástup byl ovlivněn především růstem výpočetního výkonu a s ním souvisejícím zlevňováním a zmenšováním počítačů. Protože VR zobrazovací zařízení jsou na výpočetní výkon velmi náročná, větší rozšíření mohlo přijít až se snížením cen potřebných komponent.

Nyní VR nachází čím dál větší uplatnění všude tam, kde je potřeba testovat složitější konstrukční celky, navrhovat pracovní prostředí či vyhodnocovat různé varianty návrhů. Výhodou je také relativně jednoduchá implementace 3D modelů v různých formátech od různých dodavatelů a jejich sestavování do výsledného celku, což usnadňuje aktualizace celkového modelu a efektivně zabraňuje konstrukčním kolizím. Návrh robotických pracovišť programování robotů offline a dost možná i online patří nepochybně mezi obory, kde použití VR může umožnit úsporu času a prostředků.[35]



Obr. 33: Headset pro zobrazení virtuální reality od firmy Oculus [36]

Effektu virtuální reality je možné docílit několika různými způsoby. Pravděpodobně nerozšířenější jsou brýle a headsety pro virtuální realitu, obsahující zobrazovací jednotku ve formě displeje, která zobrazuje každému oku mírně odlišný obraz, čímž se dosahuje požadovaného 3D efektu. Tato zařízení mohou být připojena ve

formě periferie k počítači, který zajišťuje výpočetní výkon a na kterém je spuštěn ovládací program. Existují také brýle, které mají výpočetní a ovládací jednotku přímo integrovanou a mohou tedy fungovat samy o sobě bez různých drátů a počítače. Jako řídicí jednotka se používají čipsety z mobilních telefonů. Tato samotná zařízení se označují jako standalone. Spolu s brýlemi jsou většinou využívány také ruční ovladače, pomocí kterých operátor s prostředím interaguje.



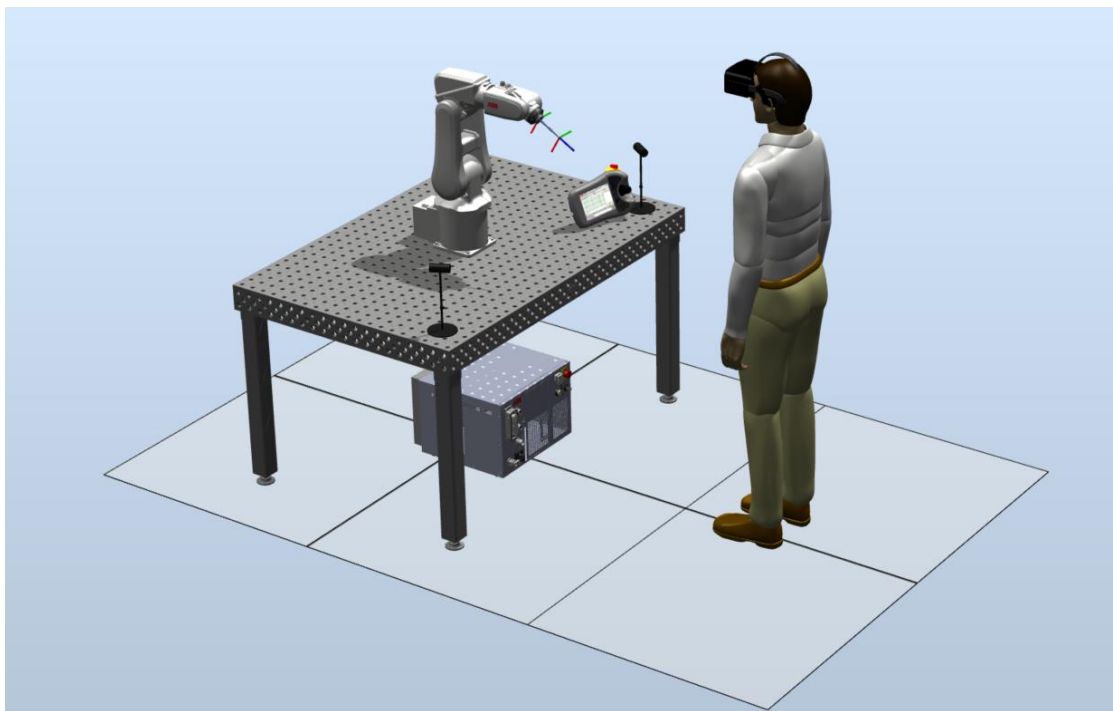
Obr. 34: Ukázka simulačního prostoru 3D Cave [37]

Další možností docílení efektu virtuální reality je použití jednoduchých pasivních brýlí a promítacího plátna tak, jak se běžně děje při 3D projekcích v kinech. Nevýhodou je velmi omezené zorné pole. Tento způsob výrazně vylepšuje přidání většího množství promítacích ploch a jejich spojení. Tento systém se nazývá 3D Cave. K projekci většinou dochází na tři stěny a podlahu, velkou výhodou je, že simulace se může účastnit více lidí zároveň a že v 3D simulovaném prostoru se lze do jisté míry pohybovat.

5.2 Ukázka propojení VR s programem RobotStudio

Vývojové prostředí RobotStudio umožňuje implementaci doplňku, díky kterému lze program propojit s VR brýlemi. V případě školní laboratoře posloužil k tomuto účelu VR headset Oculus Rift od společnosti Oculus. Spolu s brýlemi jsou v sestavě dva ovladače, každý pro jednu ruku a sledovací senzory, které rozpoznávají polohu nasazených brýlí v prostoru. Díky těmto sensorům se lze pohybovat po virtuálním prostředí prostým pohybem po laboratoři. Simulační prostředí vytvořené v programu RobotStudio

k provedení testů je zobrazeno na Obr. 35. Součástí prostředí je na pracovním stole umístěný robot IRB 120, řídicí systém robota IRC5C, FlexPendant a dvojice sledovacích senzorů.



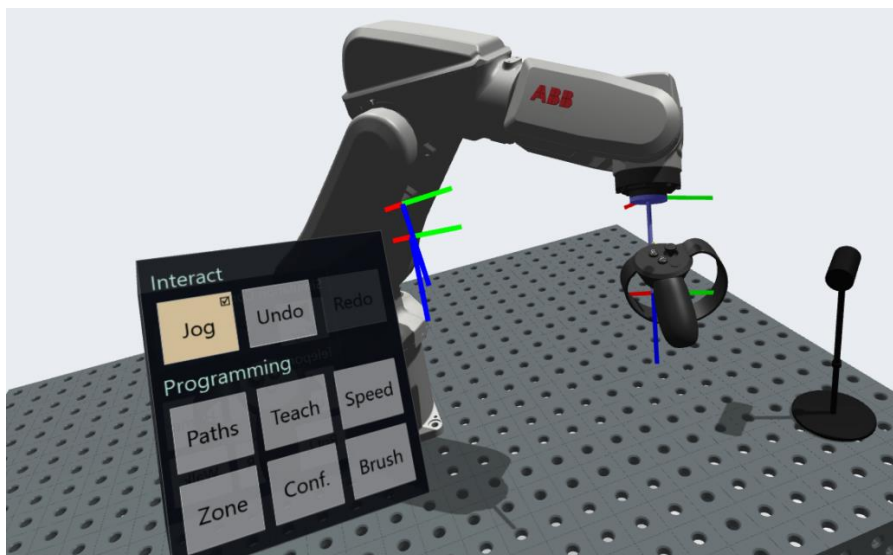
Obr. 35: Simulační prostředí vytvořené pro potřeby testu virtuální reality

Po vytvoření prostředí bylo přistoupeno k samotnému testu. Po vstupu do simulace byly postupně zkoušeny funkcionality ovládání VR prostředí. To, jak vidí prostředí osoba s nasazeným VR headsetem, je zobrazeno na Obr. 36.



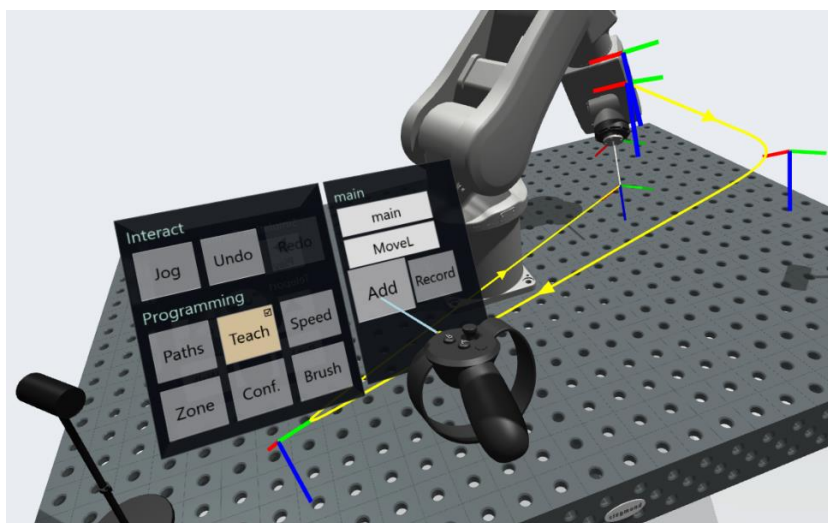
Obr. 36: Celkový pohled na prostředí tak, jak jej vidí osoba s nasazeným headsetem

V pohledu lze vidět virtuální reprezentaci obou ovladačů, které má uživatel v rukou. Ovladač v pravé ruce vypadá stejně jako v reálném prostředí a umožňuje ukazovat na jednotlivé prvky simulace, potvrzovat volby apod. Ovladač v levé ruce se ve virtuálním prostředí jeví jako krychle, kterou lze pohybem tlačítek otáčet. Na každé ze stran krychle se nachází sestava kontrolních tlačítek, které může uživatel stisknout pomocí ovladače v ruce pravé. Na Obr. 36 můžeme vidět stranu s tlačítky sloužícími k pohybu v prostoru simulace (bez nutnosti pohybu uživatele fyzicky po místnosti) a tlačítka ovládání samotné simulace, spuštění, pauza a zastavení.



Obr. 37: Ruční ovládání pohybů robota uchopením jeho jednotlivých os

Následující stranu ovládací krychle můžeme vidět na Obr.37. Zde může uživatel spustit režim označený jako *Jog*, který umožňuje ruční pohyb jednotlivými segmenty robota, nebo přímo jeho nástrojem za pomoci virtuálního uchopení.



Obr. 38: Tvorba cesty pomocí funkce *Teach*

Následující tlačítka v sekci *Programming* slouží k programování robota offline, tedy k tvorbě jednotlivých pohybů, cest a nastavování parametrů těchto cest, jako je rychlost, zóna, konfigurace apod. Tvorba cesty probíhá tak, že uživatel pomocí funkce *Jog* natáčí robota požadovaným způsobem a pomocí funkce *Teach* jednotlivé polohy robota zaznamenává. Přidává tak do prostoru jednotlivé targety, jejichž spojením vzniká cesta, po které se bude robot pohybovat. Tvorba takové cesty pomocí funkce *Teach* je znázorněna na Obr. 38.

Krom tvorby cest lze ve virtuálním prostředí také dělat poznámky, šipky, popisky a kreslit volně do prostoru. Video z testování se nachází na CD v příloze B.

5.3 Možnosti dalšího vývoje a praktického použití

Programování robotů s využitím virtuální reality je dle mého názoru obor, který se bude v následujících letech dále rozvíjet. Rozšíření možností by mohla přinést implementace programování robota nejen offline ale také online. To by přinášelo zajímavý efekt, a to že operátor by viděl robota ve virtuální realitě z bezprostřední blízkosti, ale zároveň by jím nebyl ohrožen a fyzicky se nacházel v bezpečné vzdálenosti. Další úrovní tohoto přístupu by mohlo být programování a ladění robotického stanoviště online pomocí virtuální reality pracovníkem, který by se fyzicky ani nemusel nacházet poblíž, a byl by to například expert na daný specifický problém pracující ze své stálé kanceláře. Další možností ve formě prezentace projektů a porad by mohlo být sdílení jednoho virtuálního prostoru s více uživateli, kteří by tak mohli spolu mohli intuitivně kooperovat při řešení problému.

6 ZÁVĚR

V rámci bakalářské práce byla nejdříve představena společnost ABB a některé produkty její divize zaměřené na robotiku a automatizaci. Dále proběhlo seznámení se základními principy programování robotů offline s užitím programu RobotStudio a byly představeny jednotlivé klíčové funkcionality tohoto programu, umožňující návrh robotických pracovišť, potřebné k řešení této bakalářské práce. Následně byl úspěšně navržen řídicí program robota ABB IRB 120, umožňující s pomocí robota automaticky řešit logický hlavolam hanojské věže, doplněný o ovládací aplikaci s grafickým uživatelským rozhraním.

Spolu s řídicím programem bylo také navrženo simulační prostředí, na kterém lze program testovat a do programu byly implementovány prvky umožňující interakci programu s tímto prostředím, jako je skrývání nadbytečných disků, automatické uvedení úlohy do původního stavu po provedení simulace apod. Navržený program byl nejdříve úspěšně otestován počítačovou simulací v programu RobotStudio a následně nahrán do skutečného řídicího systému reálného robota ve školní laboratoři. Implementace na reálném hardwaru proběhla také úspěšně, čímž se potvrdila přesnost předem provedené simulace.

Závěr práce se věnoval představení možnosti programování robotů s využitím technologií virtuální reality, jakožto nové funkcionality programu RobotStudio. K testování bylo vytvořeno jednoduché demonstrační prostředí, v němž byly ve školní laboratoři odzkoušeny základní funkcionality programu. Z průběhu testů virtuální reality a z testů řídicího programu pro řešení úlohy hanojských věží bylo zpracováno video.

Rozšířením této práce by mohlo být přidání dalších funkcionalit do řídicího programu. Takovou funkcionalitou by například mohla být funkce umožňující variabilní zadání pozice a natočení hlavolamu v prostoru v ovládací aplikaci na FlexPendantu, nebo funkce umožňující uživateli zvolit nejen cílovou věž řešení ale také věž počáteční. Řídicí program by také bylo možné implementovat i na jiný typ robota, ať už z portfolia firmy ABB, což by pravděpodobně nebylo příliš složité, tak na robota od některé jiné společnosti. Dalším rozšířením této práce by také mohlo být zpracování úlohy pro jiný typ koncového efektoru, který by místo zamýšlené podtlakové přísavky uchopoval jednotlivé disky jiným způsobem, a to například s pomocí kleští nebo elektromagnetu.

7 POUŽITÉ INFORMAČNÍ ZDROJE

- [1] About ABB company. *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/about/our-businesses>
- [2] ABB česká republika: základní údaje. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/cz/o-nas/zakladni-udaje>
- [3] A global nerve center. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://resources.news.e.abb.com/images/2017/3/30/0/GBS_Krakow_Axis.jpg
- [4] ABB Robotics. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics>
- [5] Akvizice společnosti B&R společností ABB. In: *Abb.cz* [online]. [cit. 2019-05-15]. Dostupné z: <http://www.abb.cz/cawp/seitp202/cfa3145a17f78ba4c12580f900367e6d.aspx>
- [6] Will BR acquisition give ABB an automation boost?. In: *Pbsionthenet.net* [online]. [cit. 2019-05-15]. Dostupné z: <http://www.pbsionthenet.net/article/133035/Will-B-R-acquisition-give-ABB-an-industrial-automation-boost.aspx>
- [7] Global market shares for for ABB and B&R for key product types. In: *Pbsionthenet.net* [online]. [cit. 2019-05-15]. Dostupné z: <http://www.pbsionthenet.net/global/showimage.ashx?Type=Article&ID=134888>
- [8] ABB industrial robots. *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots>
- [9] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUUM, 2016. ISBN 978-80-214-4828-5
- [10] ABB IRB 8700. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://www07.abb.com/images/librariesprovider89/default-album/robots/IRB-8700/irb-8700-and-person.jpg?sfvrsn=1>
- [11] Yumi manufacturing sockets. In: *Ifr.org* [online]. [cit. 2019-05-15]. Dostupné z: <https://ifr.org/news/yumi-manufacturing-sockets-at-abbs-plant-in-the-czech-republic>
- [12] ABB Yumi colaborative robot. In: *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi>
- [13] ABB IRB 14000. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://www07.abb.com/api/ir/getimage/36fb710b-54e0-4e53-8383-aacde553ec56/1>
- [14] ABB paralel robots. In: *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-360>
- [15] FlexPendant. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://library.e.abb.com/public/61d9466529af41f4b5d27514067d3211/bosch%20weldtmer%20interfaceROB0069EN_C.pdf
- [15] IRB 360 FlexPicker datasheed. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://search-ext.abb.com/library/Download.aspx?DocumentID=ROB0082EN_G&Language Code=en&DocumentPartId=&Action=Launch
- [16] ABB FlexPicker 360. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: [http://www02.abb.com/global/seabb/seabb364.nsf/bf177942f19f4a98c1257148003b7a0a/fac05bdf385c56f3c12574510025ff98/\\$FILE/IRB+360+FlexPicker.jpg](http://www02.abb.com/global/seabb/seabb364.nsf/bf177942f19f4a98c1257148003b7a0a/fac05bdf385c56f3c12574510025ff98/$FILE/IRB+360+FlexPicker.jpg)
- [17] ABB SCARA robots. In: *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-910sc>

- [18] SCARA Robot. In: *Automationworld.com* [online]. [cit. 2019-05-15]. Dostupné z: https://www.automationworld.com/sites/default/files/field/image/1602np_abb_0.png
- [19] IRB 52 painting robot. In: *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-52>
- [20] IRB 52. In: *Abcloud.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://abcloud.blob.core.windows.net/public/images/3ebe9cd0-6317-4a96-bf29-0b58f6c30e7c/presentation.jpg>
- [21] IRB 120. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://www07.abb.com/images/librariesprovider89/default-album/irb1100_2x1.jpg?sfvrsn=e2c0b814_1
- [22] IRB 120 data sheet. In: *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://search-ext.abb.com/library/Download.aspx?DocumentID=ROBO149EN_D&LanguageCode=en&DocumentPartId=2&Action=Launch
- [23] IRB 120 product specification. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=3HAC035960-001&LanguageCode=en&DocumentPartId=&Action=Launch>
- [24] Robot ABB IRB 120. *New.abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/industrial-robots/irb-120>
- [25] Industrial robot controllers datasheet. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch>
- [26] ABB Controllers. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/controllers>
- [27] FlexPendant application manual. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://library.e.abb.com/public/b60548090c886120c1257b5f002f8250/3HAC036958-en.pdf>
- [28] FlexPendant. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://library.e.abb.com/public/61d9466529af41f4b5d27514067d3211/bosch%20weldtimmer%20interfaceROB0069EN_C.pdf
- [29] RobotStudio overview. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: <https://new.abb.com/products/robotics/robotstudio>
- [30] Rapid instructions, functions and datatypes. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf
- [31] ScreenMaker application manual. In: *Abb.com* [online]. [cit. 2019-05-15]. Dostupné z: https://library.e.abb.com/public/b62731a5b2d5528bc125766d003a6228/Application_Manual_ScreenMaker.pdf
- [32] Tower of Hanoi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-15]. Dostupné z: https://en.wikipedia.org/wiki/Tower_of_Hanoi
- [33] Tower of Hanoi picture. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-05-15]. Dostupné z: https://commons.wikimedia.org/wiki/File:Tower_of_Hanoi.jpeg#file

- [34] Hanoi tower algorithm analyze. In: *Freecodecamp.org* [online]. [cit. 2019-05-15]. Dostupné z: <https://medium.freecodecamp.org/analyzing-the-algorithm-to-solve-the-tower-of-hanoi-problem-686685f032e3>
- [35] Virtuální realita v průmyslu? Již se používá. In: *Techedu.cz* [online]. [cit. 2019-05-16]. Dostupné z: <https://techedu.cz/clanky/59/virtualni-realita-v-prumyslu-uz-se-vyuziva>
- [36] Oculus Rift VR headset. In: *Roadtovr.com* [online]. [cit. 2019-05-16]. Dostupné z: <https://www.roadtovr.com/best-free-games-oculus-rift/>
- [37] Introducing 3DCave. In: *Archpaper.com* [online]. [cit. 2019-05-16]. Dostupné z: <https://cdn.archpaper.com/wp-content/uploads/2016/07/dsc03739-970x647-c.jpg>

8 SEZNAM OBRÁZKŮ A TABULEK

8.1 Seznam tabulek

Tab.1: Srovnání rychlostí jednotlivých os robotů IRB120 a IRB 120T [22]	22
Tab. 2: Srovnání parametrů řídicích jednotek IRC5 a IRC5C [25].....	24
Tab. 3: Rozměry jednotlivých disků v úloze.....	35
Tab. 4: Hodnoty stavových matic při spuštění úlohy s pěti disky.....	38
Tab. 5: Stránky ovládací aplikace.....	41
Tab. 6: Doba řešení úlohy.....	43

8.2 Seznam obrázků

Obr. 1: Hlavní obchodní centrum ABB v Krakově, Polsko [3].....	17
Obr. 2: Společný tržní podíl po akvizici B&R ve vybraných odvětvích.....	18
Obr. 3: Univerzální robot IRB 8700 v porovnání s člověkem [10].....	19
Obr. 4: Kolaborativní robot IRB 14000 YuMi [13].....	20
Obr. 5: Dvojce robotů IRB 360 FlexPicker balící konzervy [16].....	20
Obr. 6: Robot IRB 910SC s délkou ramene 550 mm [18].....	21
Obr. 7: Lakovací robot IRB 52 určený k lakování dílů na výrobních linkách [20].....	21
Obr. 8: Robot ABB IRB 120 na produktové fotografii [21].....	22
Obr. 9: Robot IRB 120 s vyznačenými osami [23].....	23
Obr. 10: Náčrt pracovního prostoru robota IRB 120 [23].....	23
Obr. 11: Řídicí jednotka IRC5C [25].....	24
Obr. 12: FlexPendant [28].....	25
Obr. 13: Prostředí programu RobotStudio s vloženým modelem robota IRB 120 a spuštěnou virtuální řídicí jednotkou.....	26
Obr. 14: Ukázka dialogového okna při tvorbě objektu tvaru válce.....	27
Obr. 15: Úvodní dialogové okno při tvorbě nového nástroje.....	28
Obr. 16: Cesta tvořená dvěma pohyby mezi třemi targety.....	28
Obr. 17: Ukázka grafického znázornění komponent propojených signály.....	29
Obr. 18: Ukázka syntaxe RAPIDu.....	30
Obr. 19: Ukázka pracovního prostředí nástroje ScreenMaker.....	31
Obr. 20: Schéma struktury GUI aplikace pro FlexPendant [31].....	32
Obr. 21: Dřevěný hlavolam hanojské věže s osmi disky [33].....	33
Obr. 22: Pohled na model hlavolamu hanojské věže ve výchozí pozici ve verzi bez středících kolíků.....	36
Obr. 23: Model efektoru osazený na koncové přírubě robota a fotografie efektoru po tisku.....	36
Obr. 24: Zjednodušený vývojový diagram řídicího programu.....	37

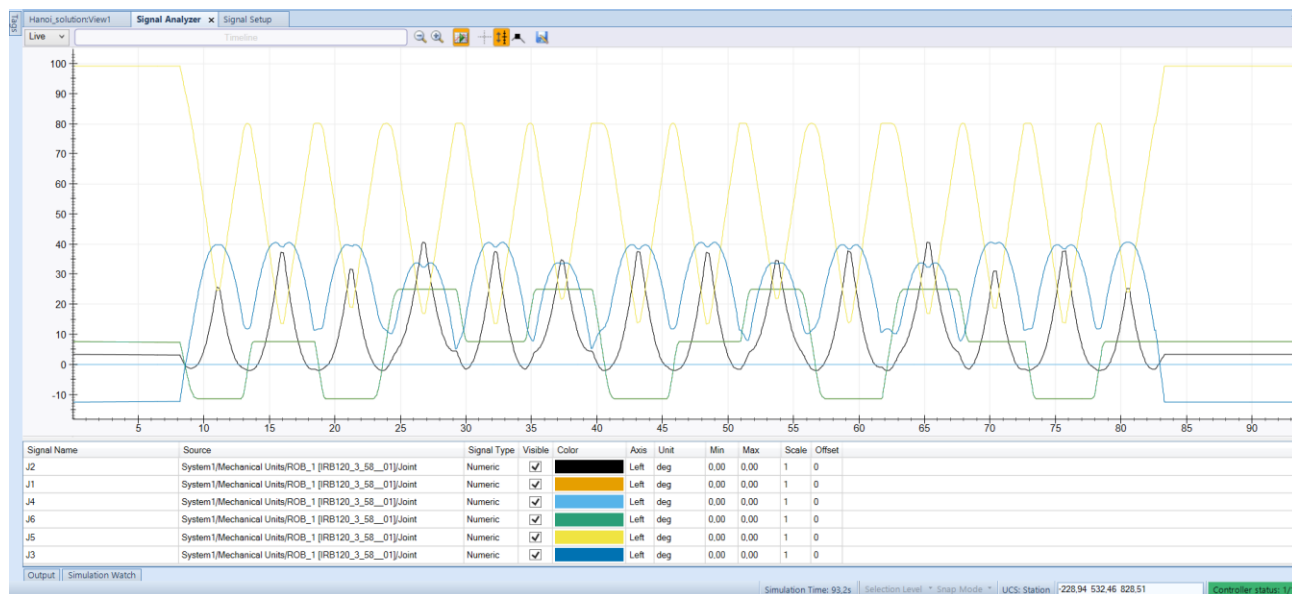
Obr. 25: Diagram procedury tahu AB.....	38
Obr. 26: StationLogic navrženého programu.....	39
Obr. 27: Úvodní stránka ovládací aplikace spuštěná na virtuálním FlexPendantu.....	40
Obr. 28: Stránka ovládací aplikace sloužící k nastavení základních parametrů úlohy....	41
Obr. 29: Stránka ovládací aplikace sloužící k ovládání běhu laboratorní úlohy.....	42
Obr. 30: Simulační prostředí v průběhu testu úlohy s pěti disky.....	43
Obr. 31: Doba řešení úlohy v závislosti na počtu řešených disků.....	44
Obr. 32: Testovací stanice ve školní laboratoři.....	45
Obr. 33: Headset pro zobrazení virtuální reality od firmy Oculus [36].....	46
Obr. 34: Ukázka simulačního prostoru 3D Cave [37].....	47
Obr. 35: Simulační prostředí vytvořené pro potřeby testu virtuální reality.....	48
Obr. 36: Celkový pohled na prostředí tak, jak jej vidí osoba s nasazeným headsetem...	48
Obr. 37: Ruční ovládání pohybů robota uchopením jeho jednotlivých os.....	49
Obr. 38: Tvorba cesty pomocí funkce Teach.....	49

9 SEZNAM PŘÍLOH

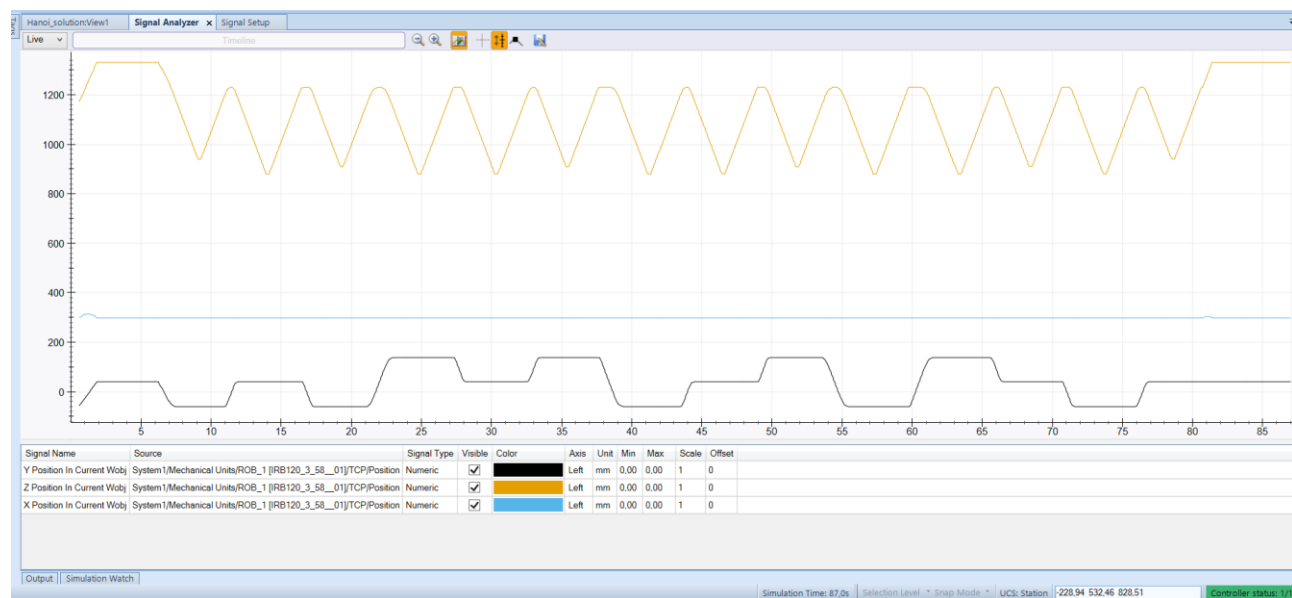
Příloha A. Grafy s údaji naměřenými v průběhu simulace

Příloha B. CD-ROM

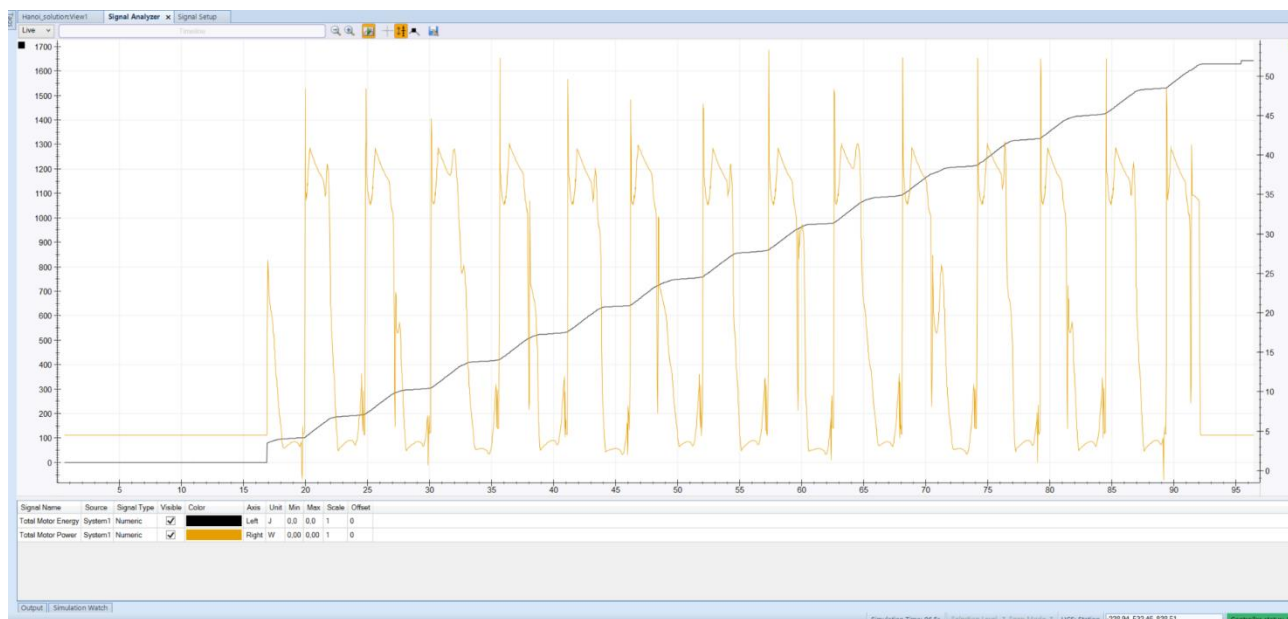
A. GRAFY S ÚDAJI NAMĚŘENÝMI V PRŮBĚHU SIMULACE



Obr.1: Natočení jednotlivých os robota v průběhu řešení úlohy se třemi disky



Obr.2: Průběh pozice efektoru robota v průběhu řešení úlohy se třemi disky



Obr.3: Příkon robota a celková spotřebovaná energie v průběhu řešení úlohy se třemi disky

B. CD-ROM

Tab. 1: Obsah CD – hlavní adresář

Adresář	Soubor
Bakalářská práce	2019_BP_Nevriva_Vaclav_192010.pdf
Úloha hanojské věže	Hanoi_solution.exe
	Hanoi_solution.rspag
	Ridici_program.txt
Úloha testování VR technologií	VR_solution.exe
	VR_solution.rspag
Video	Video_z_testovani.mp4

