



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Software pro sběr technologických dat robota KUKA VKRC4 pro účely procesní optimalizace

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

Autor práce: **Bc. Jindřich Titlbach**

Vedoucí práce: Ing. Tomáš Martinec, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jindřich Titlbach**
Osobní číslo: **M15000205**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Mechatronika**
Název tématu: **Software pro sběr technologických dat robota KUKA VKRC4 pro účely procesní optimalizace**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznámte se s roboty KUKA s řídicím systémem VKRC4 a dále standardy pro tvorbu robotických aplikací v rámci koncernu VW.
2. Vytvořte clientský software pro sběr technologických dat z robota a dále navrhňte a realizujte vizualizační software pro prezentaci naměřených dat.
3. Navrhňte a popište způsoby využití nově vytvořeného software pro optimalizaci robotických aplikací, ukažte alespoň jeden praktický případ použití tohoto software.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **KUKA System Software 8.3, Návod k obsluze a programování pro konečné uživatele, KUKA Roboter GmbH,**
- [2] **Dokumentace k řídicímu systému VKRC4**

Vedoucí diplomové práce: **Ing. Tomáš Martinec, Ph.D.**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce: **Ing. Jan Čejka**

Škoda auto a.s.

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **14. května 2018**

prof. Ing. Zdeněk Pliva, Ph.D.
děkan



Kolář
doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2017

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 10.5.2018

Podpis: Třilbach Jindřich

Poděkování

Na tomto místě bych chtěl poděkovat všem, kteří mi při zpracování této diplomové práce pomáhali nebo mě jakkoliv podporovali. V první řadě děkuji konzultantovi práce Ing. Janu Čejkovi za jeho odborný dohled, poskytnutý čas a ochotu při zpracování této práce. Dále děkuji vedoucímu práce Ing. Tomáši Martincovi, Ph.D. za jeho vedení a cenné rady. Na závěr bych chtěl samozřejmě poděkovat rodině za jejich podporu.

Abstrakt

Tato diplomová práce se zabývá vývojem SW pro sběr technologických dat z robota KUKA. Aplikace bude primárně sloužit jako diagnostika chybového stavu v komunikaci mezi průmyslovým robotem Kuka a technologií v podniku Škoda Auto v Mladé Boleslavi. Jedná se o testování komunikace signálů, tj. měření vstupů, výstupů a flagů. Lze jím ověřit správnou komunikaci robota s technologií podle požadovaného schématu komunikace, tzv. impuls-plánu. Může se jednat např. o uvolňování nebo polohování kleští či zapnutí regulátoru. Hlavní přínos je předejít možným mechanickým poruchám, např. poškození konektorů mezi dokovací spojkou a nástrojem či prasknutí greiferu pomocí získaných (naměřených) dat. S tímto souvisí měření veličin v průběhu aktuálně vykonávaného procesu robota, především měření proudu a polohy, popř. rychlosti.

Práce obsahuje teoretický úvod do problematiky průmyslu 4.0, který zobrazuje revoluci v průmyslu automatizace a robotizace. Dále se práce zabývá popisem zmíněného robota KUKA a robotickými standardy VW. V návaznosti na samotnou aplikaci je implementována přímá úloha kinematiky, která slouží k vypočtení aktuální polohy robota pomocí Eulerových úhlů, a to ze zadaných parametrů DH tabulky. Tato volba je v aplikaci jako doplněk v návaznosti na měření polohy a slouží k jejímu ověření, popř. ušetření času měření. V závěru práce jsou řešeny způsoby využití optimalizace SW.

Klíčová slova

Robotizace

Průmysl 4.0

Diagnostika

KUKA VKRC4

Měření proudu a polohy

Dokovací spojka

Abstract

This diploma thesis deals with the development of SW for the collection of technological data from Kuka robot. The application will primarily serve as a diagnostics of the error states in communication between a Kuka robot and the technology at Škoda Auto in Mladá Boleslav. It involves the testing of signal communication, ie. the measurement of inputs, outputs and flags. It is possible to verify the proper communication of the robot with the technology according to the required communication scheme, the impulse-plan. For instance, it can refer to releasing or positioning of the welding pliers or turning on the regulator. The main asset is to preempt possible mechanic malfunctions, such as damage of connectors between a docking clutch and a tool or a burst of a greifer with the help of acquired data. This is related to the measurement of quantities during the current process of the robot, especially the measurement of the current and position, eventually speed.

The thesis contains a theoretical introduction to Industry 4.0, which illustrates the revolution in the automation and robotization. In addition, the thesis deals with the description of the Kuka robot and the VW robotic standards. In connection with the application itself, a forward kinematics is used to calculate the current position of the robot using the Euler angles, from the specified DH table parameters . This option is available in the app as a supplement in connection with the measurement of the position and serves to verify it, eventually save time of the measurement. At the end of the thesis, the ways of using SW optimization are discussed.

Key words

Robotization

Industry 4.0

Diagnostics

KUKA VKRC4

The current & position measurement

A docking clutch

Obsah

Úvod	12
1 Úvod do robotizace aneb průmysl 4.0	14
1.1 Historie průmyslových revolucí	14
1.2 Charakteristika konceptu čtvrté průmyslové revoluce	16
1.3 Současný stav	17
1.3.1 Systémová integrace	17
1.3.2 Datová úložiště a cloudové služby	17
1.3.3 Autonomní roboty.....	18
1.4 Výhody a nevýhody průmyslu 4.0	18
1.5 Budoucnost průmyslu 4.0	18
2 Popis robota KUKA VKRC4 s robotickými standardy VW.....	20
2.1 Vybrané technické parametry robota KUKA KR 210 R2700 extra	21
2.2 Popis robotického systému	21
2.3 Popis manipulátoru.....	22
2.3.1 Zápěstí	23
2.3.2 Spodní rameno.....	23
2.3.3 Vyvažovací systém	23
2.3.4 Elektroinstalace.....	24
2.3.5 Základna	24
2.3.6 Vrchní rameno (spojovací).....	24
2.4 Standardizace v rámci koncernu VW	24
3 Kinematika robota a Denavit-Hartenbergovy metody.....	26
3.1 Obecný algoritmus a odvození DH metody.....	27
3.2 Přímá úloha kinematiky ve spojení s DH metodou.....	28
3.3 Implementace přímé úlohy robota.....	30
4 Vývoj SW pro sběr technologických dat z robota.....	33

4.1	Struktura programu	33
4.1.1	Realizace navázání komunikace TCP	34
4.1.2	Metoda pro čtení dat z robota	34
4.1.3	Metoda pro ukládání do souboru	36
4.1.4	Metoda pro nastavení vzorkovací periody	37
4.2	Popis aplikace	38
4.2.1	Hlavní okno „Main“	39
4.2.2	Okno Forward kinematic	41
5	Způsoby využití SW pro optimalizaci robotických aplikací.....	42
5.1	Poškození konektorů v dokovací spojce vlivem vysokého tlaku	43
5.2	Diagnostika signálů	44
5.3	Praskání greiferu vlivem přetížení	47
5.4	Další způsoby využití	48
6	Návrhy na vylepšení SW	49
7	Závěr	50
	Seznam použité literatury	51
	A Příloha – Obsah přiloženého CD.....	54
	B Příloha – Metoda pro přepočet na aktuální proud	55
	C.1 Příloha – Standard topologie pro zasítování	56
	C.2 Příloha – Standard pro označování profinetu	57

Seznam obrázků

Obrázek 1: Parní stroj [1].....	14
Obrázek 2: Teslův transformátor v Australském národním muzeu [2]	15
Obrázek 3: Diagram průmyslových revolucí	15
Obrázek 4: Robot KUKA KR 210 R2700 EXTRA.....	20
Obrázek 5: Robotický systém [8].....	22
Obrázek 6: Sestava manipulátoru.....	23
Obrázek 7: Kinematika DH [11]	27
Obrázek 8: Angulární kinematická struktura robota	29
Obrázek 9: Kartézské souřadnice chapadla	30
Obrázek 10: Připojení kabelu do sběrnice	33
Obrázek 11: Formát pro přenos dat [20].....	35
Obrázek 12: Blokové schéma komunikace	36
Obrázek 13: Struktura souboru.....	37
Obrázek 14: Hlavní okno aplikace.....	39
Obrázek 15: Ukázka grafu	40
Obrázek 16: Okno Forward kinematic.....	41
Obrázek 17: Ukázka vypočtených hodnot	41
Obrázek 18: Trénovací buňka	42
Obrázek 19: Dokovací spojka s křídélkem	43
Obrázek 20: RIP zařízení a odsávací válec.....	43
Obrázek 21: Ukázka z impuls-plánu (rozměry otevření kleští).....	45
Obrázek 22: Schéma externí automatiky	46
Obrázek 23: Odezva výstupů Folge	47
Obrázek 24: Prasknutý greifer [21]	47
Obrázek 25: Výsledky napětí metodou Von Mises [22].....	48

Seznam tabulek

<i>Tabulka 1: Tabulka výhod/nevýhod průmyslu 4.0</i>	<i>18</i>
<i>Tabulka 2: Vybrané technické parametry KR 210 R2700 extra</i>	<i>21</i>
<i>Tabulka 3: Vybrané standardizované prvky</i>	<i>25</i>
<i>Tabulka 4: Tabulka DH parametrů.....</i>	<i>28</i>
<i>Tabulka 5: Aktuální hodnoty otevření kleští.....</i>	<i>46</i>
<i>Tabulka 6: Dopady prasklého greifery ve výrobě</i>	<i>48</i>

Seznam zdrojových kódů

<i>Zdrojový kód 1: Ukázka kódu pro výpočet Eulerových úhlů z matice rotace</i>	<i>32</i>
<i>Zdrojový kód 2: Metoda pro připojení k hostiteli.....</i>	<i>34</i>
<i>Zdrojový kód 3: Metoda readVar.....</i>	<i>35</i>
<i>Zdrojový kód 4: Část kódu pro zápis dat do souboru</i>	<i>37</i>
<i>Zdrojový kód 5: Metoda Timer.....</i>	<i>38</i>

Seznam grafů

<i>Graf 1: Průběh nadokování kleští.....</i>	<i>44</i>
<i>Graf 2: Průběh otevírání a zavírání kleští</i>	<i>45</i>

Seznam zkratek

PLC	Programmable Logic Controller
ČSÚ	Český statistický úřad
EDI	Electronic Data Interchange
RFID	Radio Frequency Identification
KCP	Kuka control panel
KRC	Kuka robot controller
MFEE	Multi-function end effector
RDC Box	Resolver digital converter
MFH	Multi-function housing
ESC	Electronic safety circuit
DH	Denavit-Hartenberg
UTP	Unshielded Twisted Pair
TCP	Transmission Control Protocol
PC	Personal computer
RIP	Robot installation plate
ISO	International Organization for Standardization

Úvod

Průmyslového robota lze označit za automatický stroj, který obsahuje manipulátor s minimálně dvěma osami a programovatelný řídicí systém pro vykonávání pohybových a řídicích funkcí ve výrobním procesu. Tyto funkce nahrazují analogické funkce člověka při manipulaci předmětů či jiných technologických nástrojů. První průmyslový robot je datován k první polovině 20. století. Praktické využití zaznamenali roboti značek Versatran a Unimate až o několik let později v oblasti svařování. První model průmyslového robota byl vyvinut švédskou firmou Asea v roce 1972 a nesl název IRB6. Robot disponoval 5 osami a nosností 6 kg. V tuzemsku byl vyvinut první průmyslový robot (PR 32E) až v roce 1981, který byl navržen pro svařování. [12]

Úvodní část diplomové práce je věnována velmi diskutovanému názvu „průmysl 4.0.“ Robot je především spojen s pojmy automatizace a digitalizace. Tyto pojmy v současné době ovlivňují trh práce. Automatizace snižuje potřebu přítomnosti člověka při vykonávání dané práce. Výhodou je efektivita a bezpečnost technologických procesů. S tímto pojmem významně souvisí i robotizace, ke které dochází především v automobilovém průmyslu, např. Škoda Auto, ale i v dalších odvětvích a má za cíl zvýšit produktivitu práce. Digitalizací lze rozumět např. realizaci digitální továrny, která se zabývá plánováním, ověřováním výrobních procesů a simulací výroby. V oblasti plánování svařovny ve Škoda Auto se využívá především k simulaci robotů, která slouží k návrhu robotizovaných pracovišť. Ve virtuálním prostředí, např. v programu „Process Simulate“ lze sestavit 3D model robotizovaného pracoviště nebo celé robotizované výrobní linky. Tento vytvořený 3D model lze využít pro ověřování a simulaci v různém prostoru a času. K jeho hlavním přínosům patří okamžitá zpětná vazba mezi návrhem robotizovaného pracoviště a konstrukčním návrhem nástrojů. Dále také možnost nalezení optimálního vzájemného umístění robotů a zařízení na pracovišti, včetně detekce případných kolizí. Velká výhoda digitalizace je online přístup k informacím a datům, což s sebou nese zefektivnění a zlepšování fungování procesů ve firmách. V budoucnosti vzniknou nové profese, které zatím není možno zcela specifikovat. Mnohé současné profese budou nahrazeny právě roboty, což se již děje. Jedna z nepřilíš realistických myšlenek mnoha lidí na budoucí vývoj je, že stroje ovládnou svět a člověk ztratí ve světě využití. Toto tvrzení je spíše považováno za utopii. S jistotou lze ale říci, že největší uplatnění na trhu budou mít lidé z oblasti IT a souvisejících technických oborů.

Tato diplomová práce se zabývá robotikou a digitalizací. Cílem je vytvořit software na sběr technologických dat robota KUKA VKRC4 pro účely procesní optimalizace. Sběr technologických dat robota umožní studovat procesy do větší hloubky a hledat možná zlepšení a nápravná opatření. Ze strany výrobce robotů KUKA není v současné době k dispozici jednoduchý nástroj, který by se tímto problémem zabýval a který by vyhovoval požadavkům firmy Škoda Auto.

První část práce je věnována seznámení se s robotem KUKA KR210 R2700 EXTRA C4 FLR a robotickými standardy VW. Mechanika KR210 R2700 EXTRA je hojně využívána pro projekty ve svařovnách firmy Škoda Auto. Uvádí se zde její popis, vybrané parametry a vlastnosti. V další části je řešena kinematika tohoto robota. Je zde obsažen obecný algoritmus a odvození Denavit-Hartenbergovi metody, řešení přímé úlohy kinematiky ve spojení s Denavit-Hartenbergovou metodou a následná implementace přímé úlohy kinematiky robota. Dále je uveden stručný vhled do programovacího jazyku C#.

V hlavní části práce je řešena struktura programu, např. komunikace mezi robotem a PC a některé významné metody. V návaznosti na strukturu je uveden popis aplikace, resp. software pro optimalizaci robotických aplikací, jehož hlavní součástí je sběr technologických dat, která jsou ukládána do souboru a zobrazována pomocí grafu.

Závěrečná část práce se zabývá způsoby využití softwaru včetně konkrétních praktických příkladů ze svařoven firmy Škoda Auto.

1 Úvod do robotizace aneb průmysl 4.0

Tato diplomová práce se zařazuje do oblasti robotizace, která je velmi blízce spojena právě s automatizací a digitalizací, navazujícími na průmysl 4.0. Co konkrétně jednotlivé pojmy znamenají, bylo řečeno již v úvodu. Průmysl 4.0 označuje aktuální trend digitalizace a další oblasti, který s tímto oborem souvisí, tj. automatizace výroby. Dále bude uvedeno, odkud tento název vzniknul, za jakým účelem a proč je označován jako čtvrtá průmyslová revoluce.

1.1 Historie průmyslových revolucí

První průmyslová revoluce se odehrála ke konci 18. století, kdy práci člověka začaly z větší části přebírat stroje. Hlavní roli v této době sehrál parní stroj. Prudký nárůst mechanizace do průmyslu a zemědělství vedl k masivnímu růstu produktivity, který následoval historicky výrazným růstem populace, kulturním a společenským změnám. Tento proces je nazýván jako industrializace. Průmyslová revoluce probíhala ještě v 19. století, kdy se dokončoval přechod od ruční výroby v manufakturách ke strojní velkovýrobě. Významem by se dal přirovnat s neolitickou revolucí, kde proběhla proměna společnosti od lovců a sběračů k zemědělské (viz *Obrázek 1*). [1]



Obrázek 1: Parní stroj [1]

Druhá průmyslová revoluce přišla přibližně o 100 let později, kdy nastalo období elektrifikace a motorizace. Elektrifikaci lze vysvětlit jako proces úpravy nebo přeměny systému tak, aby ke své činnosti využíval elektrickou energii, např. elektrifikace železničních tratí. Tento pojem označuje zejména výstavbu infrastruktury pro zásobování domů a továren elektrickou energií, ke které docházelo především na přelomu 19. až 20. století. Mezi největšími vynálezy této doby byla elektrická žárovka, kterou zdokonalil

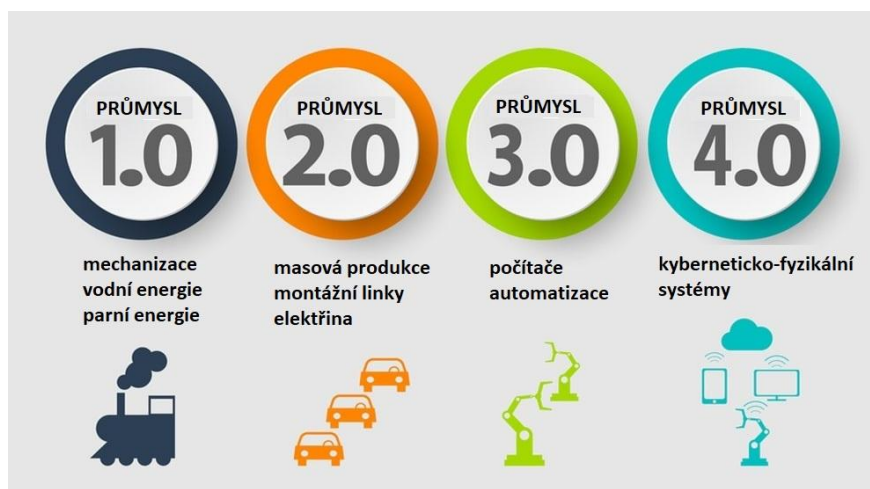
a jako první začal využívat T. A. Edison v roce 1879. Dále to byl vysokofrekvenční transformátor vynalezený Nikolou Teslou (viz *Obrázek 2*) nebo spalovací motor, o který se zasloužil G. Daimler. [3]



Obrázek 2: Teslův transformátor v Australském národním muzeu [2]

Třetí průmyslová revoluce začala na konci 60. let dvacátého století s nástupem výpočetní techniky, která přinesla automatizaci řady odvětví lidské činnosti a prudce zrychlila technický vývoj. Bývá často spojována s automatizací, elektronikou a rozmarem informačních technologií. Právě v roce 1969 byl vyroben první programovatelný logický automat (PLC). Jedná se vlastně o malý průmyslový počítač používaný pro automatizaci procesů v reálném čase – řízení strojů nebo výrobních linek v továrnách. Charakteristickou vlastností PLC je, že se program vykonává v cyklech. [1][3]

Přehled všech období průmyslových revolucí zachycuje následující diagram (viz *Obrázek 3*).



Obrázek 3: Diagram průmyslových revolucí

1.2 Charakteristika konceptu čtvrté průmyslové revoluce

O čtvrté průmyslové revoluci se začalo hovořit teprve na počátku aktuální dekády. Koncept vychází z dokumentu, který byl představen na veletrhu v Hannoveru v roce 2013. Základní vize tzv. čtvrté průmyslové revoluce se objevily v roce 2011. Zdroje [4] uvádí, že podle této myšlenky vzniknou „inteligentní továrny“, které budou využívat kyberneticko-fyzikální systémy, které jsou založeny na elektronice, inteligentních senzorech a sítích vestavěných do fyzických systémů a procesů. Ty následně převezmou opakující se a jednoduché činnosti, které doposud vykonávali lidé. To celé bude provázet změna pracovního trhu, kde je riziko ohrožení zaměstnanosti osob, u kterých počítače s novými řídicími systémy či těmito systémy řízené robotickými systémy nahradí lidskou sílu. Příkladem mohou být diagnostické lékařské systémy nebo výrobní linky, které nahradí pracovníky s nižší kvalifikací. Již v roce 2011 v ústavu mikrobiologie Fakultní nemocnice v Olomouci se začal využívat moderní laboratorní robotický systém pro zpracování vzorků na bakteriologická a mykologická vyšetření. Přístroj nahradil některé manuální úkony laborantů a poskytl zároveň přesnější provedení vyšetření. Dříve museli analýzu vzorků provádět laboranti ručně, poté byla většina úkonů zajišťována automaticky.

Měla by vznikat nová pracovní místa, která však budou vyžadovat vyšší kvalifikaci zaměstnanců. Předpokládá se, že jejím výsledkem bude téměř naprostá automatizace výroby včetně kontrolních a řídicích procesů, které dnes ještě stále obsluhují lidé. Dle predikcí z různých zdrojů by tato revoluce měla trvat ještě minimálně 10 až 30 let.

Zavádění průmyslu 4.0 probíhá kontinuálně pomocí metod strojového vidění, auto-diagnostiky a s počítačovým propojením strojů a dílů. Produkty i stroje budou vybaveny čipy, pomocí nichž je bude možné kontrolovat a obsluhovat přes internet. Dále se využívají cloudová úložiště, 3D tisk, datová centra, automatické hlášení problémů či „chytřé sklady“, které samy informují o docházejících zásobách.

Základní charakteristiky inteligentních továren:

- optimalizace výrobních procesů v rámci celého řetězce díky integrovaným IT systémům
- izolované výrobní jednotky jsou nahrazeny plně automatizovanými a vzájemně propojenými výrobními linkami
- fyzické prototypy jsou nahrazeny virtuálními návrhy výrobků
- vzájemná komunikace mezi roboty
- automatizované logické zázemí, které využívá autonomní vozíky a roboty se automaticky přizpůsobuje potřebám výroby. [4]

1.3 Současný stav

1.3.1 Systémová integrace

Úroveň budoucího rozsahu integrace u řady podniků je stále na začátku. Statistiky ČSÚ udávají, že méně než 10 % malých podniků, přibližně 15 % středních podniků a přibližně 32 % velkých podniků realizuje komunikaci EDI (Electronic Data Interchange) v oblasti nákupu a prodeje. EDI je moderní způsob komunikace mezi dvěma nezávislými subjekty, při které dochází k výměně standardních strukturovaných obchodních a jiných dokumentů (např. objednávek, dodacích listů, faktur atd.) elektronickou formou. Obdobný stav je u využití RFID systémů. Údaje ukazují, že integrace mezipodnikové komunikace je především u malých a středních firem problematická kvůli vysokým nákladům na jejich zavedení. Při sledování skladových zásob a pohybu dílů je situace o dost horší. Tyto systémy jsou podle posledního výzkumu z roku 2014 využívány u téměř 3 % malých podniků, 13 % středních podniků a téměř 27 % velkých podniků.[4]

1.3.2 Datová úložiště a cloudové služby

Mnoho poskytovatelů nabízí tzv. geograficky oddělená datová centra dostupná po celém světě, která přináší vyšší dostupnost služeb. Důležité je schopnost rozlišit úroveň a skutečnou schopnost jednotlivých poskytovatelů dodržet určité parametry datových center a poskytovaných služeb. Jedno z možných řešení jsou standardní certifikace určená pro datová centra, které přesně definují dostupnost služeb, standardů a norem definujících bezpečnost.

Cloudové služby jsou služby využívající tzv. cloud computing, tedy model, v němž jsou servery, úložiště, služby a aplikace, které služba nabízí a jsou dostupné uživateli vzdáleně na síti, čímž nezatěžují hardware ani software zařízení, se kterým do služby vstupuje. Pro využívání mnoha cloudových služeb není potřeba instalace a uživatel se do nich přihlásí pomocí klientského prostředí. Využití cloudových řešení otevírá široké spektrum možností pro růst produktivity a optimalizace nákladů na IT ve velkých, ale i středních a malých podniků, pro které by vlastní výstavba datových center nebyla ekonomicky možná. Riziko zneužití a zcizení dat se ve světě neustále zvyšuje a tak musí být kladen velký důraz na ochranu kritických systémů neboli kybernetickou bezpečnost.

Využití cloudových služeb se podle posledních dat z ČSÚ v roce 2014 pohybovalo u malých firem okolo 14 % a u velkých firem to bylo kolem 19 %. Nejvyužívanější

službou byl email. Až 47 % lidí mělo obavy z nedostatečné bezpečnosti a stejný počet lidí také z neznalosti cloudových služeb.[4]

1.3.3 Autonomní roboty

Uplatnění robotů je především v hromadné výrobě a představují významný prostředek pro zvýšení produktivity. V současnosti jsou výrobní linky v ČR vybaveny především roboty, které jsou navrženy pro specifické úkony v rámci výrobního procesu a nemají schopnost se rozhodovat autonomně. Pokrok ve vývoji robotů, ale stále pokračuje a začínají se objevovat univerzálnější roboti. Velké firmy, např. automobilky se na robotizaci soustřeďují více, z důvodu vyšší produkce a úspory pracovních sil, např. ve firmě Škoda Auto, kde se hledají konkrétní pracoviště obsluhující člověkem, na které by bylo vhodné nasadit robota. Nicméně zavádění robotů do podniku vyžaduje vysoké počáteční investice a také nové typy profesí. Dle člena představenstva automobilky VW Horsta Neumanna stojí hodina práce německého pracovníka ve zpracovatelském průmyslu 40 eur, zatímco hodinové náklady práce robota vycházejí na 5 eur, tedy 8 krát nižší náklady robota ve srovnání s člověkem, což je velmi podstatný rozdíl. [4][5]

1.4 Výhody a nevýhody průmyslu 4.0

V tabulce (viz *Tabulka 1*) jsou uvedeny výhody a nevýhody, které s sebou přináší průmysl 4.0. Shrnutím všech kladů a záporů lze říci, že klady převažují nad záporami.

Tabulka 1: Tabulka výhod/nevýhod průmyslu 4.0

Výhody	Nevýhody
úspory času a peněz	hackerské útoky
zvýšení flexibility firem	zneužití dat
zvyšování produktivity práce	zánik řady pracovních míst
dostupnost dat	velká počáteční investice
zvýšení bezpečnosti na pracovišti	vyšší požadavky na vzdělání
rychlá návratnost investic	-
snížení materiálových ztrát	-

1.5 Budoucnost průmyslu 4.0

Dnes je velmi těžké říci, co bude za dvacet až třicet let. Jaké technologie doba přinese a jak moc se dnešní doba změní. Lidská vynalézavost totiž přesahuje představivost lidského chápání. Jednou z budoucích a realistických vizí může být například řízení automobilu bez řidiče, tzv. autonomní řízení, Nedávno byl na webu zveřejněn článek [6], že

vůz společnosti UBER ovládaný autopilotem, zabil v USA člověka při přecházení silnice. Společnost následně přerušila testování samořízených aut v USA i Kanadě. Tato nehoda zcela jistě zpomalí snahu o povolení těchto vozů v USA. Vývoji těchto vozů se věnuje již déle také společnost Google nebo výrobce elektromobilů Tesla Motors. Ukazuje se, že tyto vozidla mohou být nebezpečná a zneužitá, pokud bude systém možno vzdáleně ovládat. Nyní probíhá proces optimalizace řízení, tj. testování v různých povětrnostních podmínkách, terénech a stále zlepšováním „zraku“ jednotlivých senzorů. Podle optimistických odhadů z různých zdrojů by se první autonomní vozy mohli objevit na trhu po roce 2020, kde budou z počátku sloužit především jako robotické taxíky.

Největším zpracovatelským průmyslem v ČR je výroba motorových vozidel, u které jsou zaznamenány roční tržby 1,2 bilionů Kč. Je zde i nejnižší míra nezaměstnanosti v celé EU, což s sebou ale nese také negativní dopad na stranu podniků, kdy chybí kvalifikované i nekvalifikované profese. Právě jedním z řešení nedostatku pracovníků je a bude automatizace výroby a robotizace, tedy nasazení průmyslových robotů.

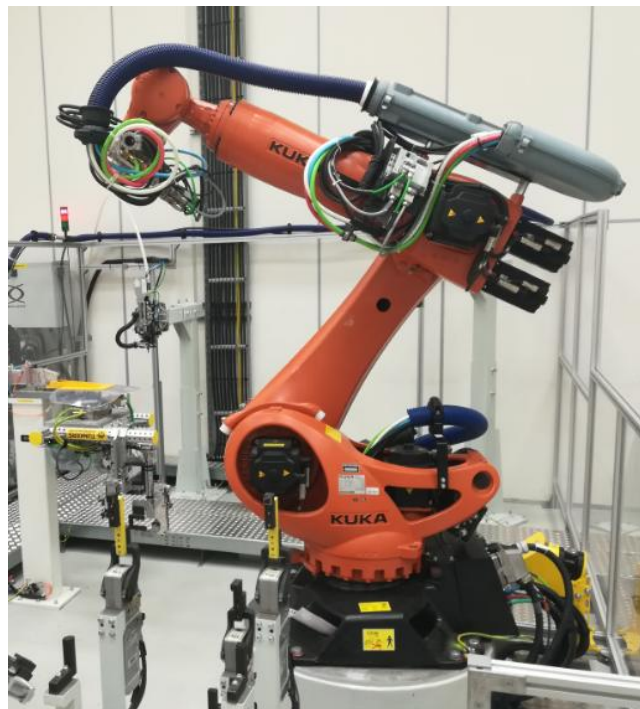
Velký potenciál je i u tzv. kooperujících robotů, kteří spolupracují v kontaktu s člověkem. Jsou flexibilní a vhodné např. pro manipulaci předmětů, šroubování nebo lepení. Zdroj [5] uvádí, že odhad na trhu s kooperujícími roboty by se měl rozrůst z původních sto milionů dolarů v roce 2014 až na více než miliardu dolarů v roce 2020. Část analytiků předpovídá, že do dvou let se stanou nejprodávanějšími zařízeními v průmyslu. Jejich cena má přitom rychle klesat na úroveň mezi 15 až 20 tisíci dolary za kus.

Rozvoj Průmyslu 4.0 souvisí také se vzděláváním. Nejen, že je vzdělávací systém jeho rozvojem ovlivňován, ale i vzdělávání má opačný vliv na jeho kvalitu. Rozvíjející se kompetence studentů může k rozvoji průmyslu výrazně přispět. Podle různých studií již dnešní výuka nestačí současným nárokům na úroveň absolventů škol. Uplatnění Průmyslu 4.0 v celé společnosti vytvoří potřebu získávání nových znalostí a dovedností, neboť důsledky Průmyslu 4.0 se budou projevovat ve všech sférách života společnosti.

[4]

2 Popis robota KUKA VKRC4 s robotickými standardy VW

Jak již bylo řečeno v úvodu, tato práce se zabývá robotem KUKA KR210 R2700 EXTRA C4 FLR (dále už jen KR 210 R2700 extra*) z řady KR QUANTEC extra (viz *Obrázek 4*). Jedná se o průmyslového robota, který nese označení „nejvšestrannějšího“ robota ze své řady. Robot je dostupný v několika provedeních s různými zatíženími. Tento robot disponuje nejvyšším zatížením z celé řady. Veškeré důležité parametry a specifikace jsou uvedeny v tabulce, viz kapitola 2.1. Jsou k dispozici dvě různá provedení, z hlediska použití. Prvním z nich se nazývá Foundry, které je vhodné pro použití robota do prašného, horkého a vlhkého prostředí. Druhé z nich jsou varianty s extra přesnými manuálními osami, které slouží pro maximální přesnost při laserových aplikacích nebo měření konstrukčních dílů. Dále je samozřejmě také možné zvolit montážní plochu, tedy podlahu nebo strop.[7]



Obrázek 4: Robot KUKA KR 210 R2700 EXTRA

Robot je zde využíván především v těchto oblastech – bodové svaření, paletizace, laserové svařování a řezání, měření a kontrolování, letování, manipulační úlohy a další montážní či demontážní aplikace. Například ve firmě DMG Mori robot vykládá a nakládá obráběcí stroje s maximální přesností. Výhody robota jsou vyladěnost na vysoké koncové rychlosti a maximální přesnost. Je zde garantována absolutní přesnost opakování polohy, díky speciálně dimenzované mechanice. Další poměrně velkou předností je

náročnost na údržbu. Podle testů [7] robot vykazuje nejdelší intervaly údržby, a to až 20 000 provozních hodin (s maximální produktivitou). Velmi dobře tak kompenzuje vznikající procesní síly, zásluhou extra stabilní struktury.

2.1 Vybrané technické parametry robota KUKA KR 210 R2700 extra

Následující *Tabulka 2* zobrazuje nejdůležitější informace o robotovi, se kterými by měl být uživatel seznámen.[8]

Tabulka 2: Vybrané technické parametry KR 210 R2700 extra

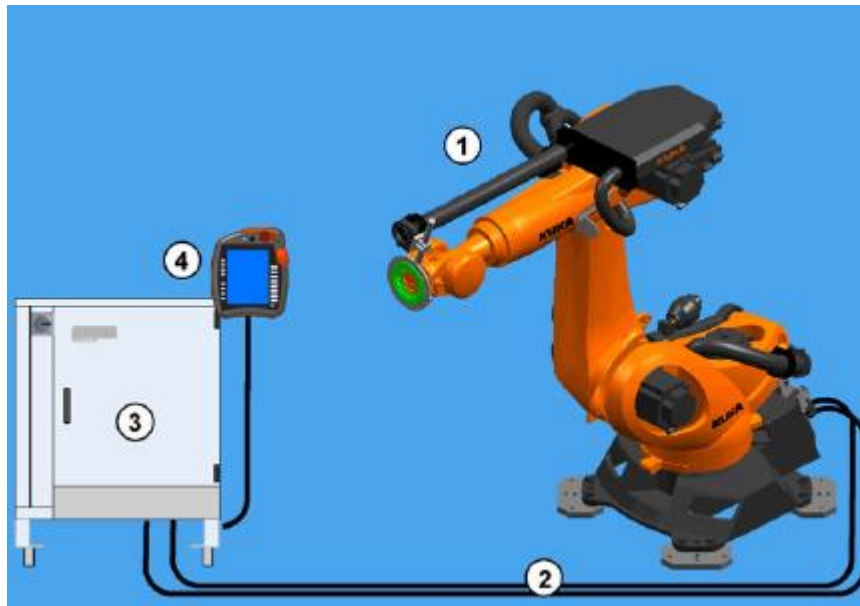
Kategorie	KR 210 R2700 extra
Počet ovladatelných os	6
Nosnost	210 kg
Hmotnost	1068 kg
Rozsah	2696 mm
Přesnost opakovaného nastavení polohy	+/- 0,06 mm
Montážní poloha	podlaha
Okolní teplota	10 °C až 55 °C
Krytí	IP 65
Řídicí systém	VKR C4
Přesný název robota*	KR210 R2700 EXTRA C4 FLR

2.2 Popis robotického systému

Robotický systém (viz *Obrázek 5*) zahrnuje všechny následující součásti průmyslového robota:

1. Manipulátor (mechanický systém a elektroinstalace)
2. Připojovací kabely
3. Řídicí systém robota
4. KUKA smartPAD (dotykové zařízení pro ovládání robota)

Dále může být obsahem software či další příslušenství, např. koncové efekторы nebo lineární jednotky. Robot může být instalován s různými možnostmi, např. jako systémy napájení pro osy 1 až 3 nebo pro osy 3 až 6, montážní přírubou (adaptér) nebo ovládacím kabelem pro jednu osu. Popis manipulátoru je uveden v následující kapitole 2.3.



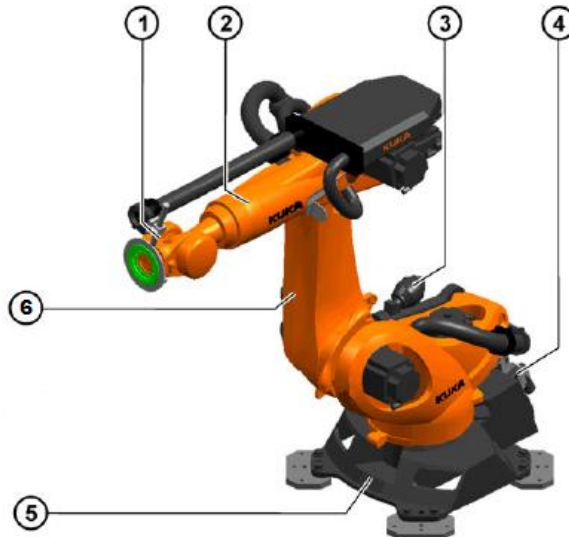
Obrázek 5: Robotický systém [8]

Řídicí systém VKRC4 je ve své podstatě běžný KRC4, s tou změnou, že je kompletně napojen na VW standard. Tato skutečnost dává omezení implementovat jakékoliv nové změny bez schválení z koncernu VW. Systém KRC4 je považován za průkopníka v automatizaci. Snižuje náklady na integraci, údržbu a péči. Díky otevřeným standardům zvyšuje efektivnost a flexibilitu systémů. Součástí KRC4 jsou řídicí systémy: Robot Control, PLC Control, Motion Control a Safety Control, což dělá tuto koncepci revoluční. Všechny tyto systémy sdílí stejnou infrastrukturu a tím je automatizace snazší a výkonnější. Je zde zajištěna např. podpora PLC, což umožňuje kompletní přístup k celému systému vstupů a výstupů robota, robotické buňce nebo robotickým linkám.

Do řídicího systému je kompletně integrováno bezpečnostní řízení. Bezpečnostní funkce a komunikace je realizována prostřednictvím protokolů na bázi ethernetu. Řídicí systém pracuje spolehlivě v různých podmínkách, jako je extrémní teplo nebo chlad či vlhkost. Je vybaven 25 jazyky a splňuje veškeré celosvětové normy ISO a standardy US. K dispozici je pět provedení řídicího systému KRC4, které se liší především rozměry a požadavky ohledně stohovatelnosti nebo ochrany před ostatními vlivy, např. prach a vlhkost. [7]

2.3 Popis manipulátoru

Pojem manipulátor lze v robotice označit za rameno robota a elektroinstalaci (viz *Obrázek 6*). Tento robot je navržen jako 6 osá kinematická struktura, která je složena z níže uvedených částí:



Obrázek 6: Sestava manipulátoru

2.3.1 Zápěstí

Robot je vybaven 3osým zápěstím (1), které je spojeno s osami 4, 5 a 6 (viz Obrázek 8) v kapitole 3.2, kde lze vidět rozložení jednotlivých os. Motor osy 6 je umístěn přímo na zápěstí uvnitř ramena, to způsobuje jeho přímý pohon. Osy 4 a 5 jsou poháněny zezadu ramene přes spojovací hřídele. Zápěstí je vybaveno montážní přírubou pro připojení koncových efektorů (nástrojů), např. multifunkčního koncového efektoru pro automatizované vrtání nebo nýtování. MFEE si dokáže zvolit a samostatně přepínat mezi vhodnými funkcemi. Ve Škodě Auto je především jako nástroj použit greifer (chapatlo) pro bodové svařování s automatickým frézováním čepiček nebo lepení. [8]

2.3.2 Spodní rameno

Spodní rameno (2) je článek mezi zápěstím a vrchním ramenem. Jsou v něm umístěny motory ze zápěstí z os 4 a 5. Rameno je poháněno motorem z osy 3. Maximální přípustný úhel natočení je mechanicky omezen zastavením pro kladný i záporný směr. K dispozici je rozhraní se 4 otvory pro upevnění dodatečné zátěže, které lze upevnit na rameno. [8]

2.3.3 Vyvažovací systém

Vyvažovací systém (3) je instalován mezi základnou a vrchním ramenem. Slouží ke snížení momentů kolem osy 2 za stavu, kdy je robot v pohybu nebo klidu. K tomu dochází za použití uzavřeného hydropneumatického systému. Vyvažovací systém je složen ze dvou akumulátorů, hydraulického válce s přidruženými hadicemi, manometru a speciálního kotouče jako bezpečnostního prvku pro ochranu proti přetížení. [8]

2.3.4 Elektroinstalace

Elektrická instalace (4) zahrnuje všechny motory a datové kabely pro motory z os 1 až 6. Všechna připojení jsou realizována jako konektory, aby bylo možné vyměnit motory rychle a spolehlivě. Elektroinstalace také zahrnuje RDC Box, který je umístěn v základně a nedílnou součástí je multifunkční skříň. MFH a konektor pro datové kabely jsou upevněny na základnu (rám) robota. Připojovací kabely od řídicí jednotky jsou zde propojeny pomocí konektorů. Součástí elektroinstalace je ochranný obvod, tedy pokud se robot pohybuje mimo rozsah bezpečnosti elektronického bezpečnostního obvodu (ESC), systém odpojí pohony. [8]

2.3.5 Základna

Rám (base frame) je základnou robota (5) a je přišroubován k podstavci robota. Je zde připevněna ohebná trubka pro elektroinstalace. Na rámu je také umístěno rozhraní pro pohonnou jednotku, datový kabel a energetický systém. [8]

2.3.6 Vrchní rameno (spojovací)

Spojovací rameno (6) je pevná část robota umístěna mezi spodním ramenem a základnou. V kombinaci se spodním ramenem je možno dosáhnout požadovaného dosahu. Stejně jako u spodního ramena je zde k dispozici rozhraní se 4 otvory pro upevnění dodatečné zátěže, které lze upevnit na spojovací rameno. [8]

2.4 Standardizace v rámci koncernu VW

Standardizace obecně znamená proces sjednocení pomocí zavádění standardů. Hlavním přínosem je koordinace, kompatibilita a opakovatelnost v kvalitě výroby. Standardy se většinou zavádějí centrálně. Pokrok v rozvoji standardů přinesla až průmyslová revoluce, např. pro specifikaci tvarů závitu. [9]

V této práci bude řešena především standardizace v oblasti svařovny. Hlavním úkolem v závodě Škoda auto je vyrábět auta, což s sebou nese tlak na cenu a rychlost výstavby zařízení. Standardizace komponent, tj. automatizačních prvků představuje svůj systém a přehlednost pro údržbu. Cílem je minimalizace rizik problémů řízení firmy, zvýšení efektivity a produktivity v podniku. Ve svařovně dochází ke standardizaci především u robotů, ale i dalších komponent či zařízení (viz *Tabulka 3*).

Tabulka 3: Vybrané standardizované prvky

Vybrané standardizované prvky v rámci VW
roboti KUKA a FANUC
rozhraní mezi PLC a ROBOTEM
E-PLAN pro vytváření projektových dokumentací
elektrická zapojení
programování (využití standardizovaných bloků a maker)
ventilové terminály
senzory
bezpečnostní skenery

V rámci VW existuje oddělení, které řeší oblast standardizace, tj. udržuje tyto komponenty a jejich aktuální stav. Výstavba nových automatizovaných linek vzniká také podle příslušného standardu, který zajišťuje identitu v rámci celého koncernu, tedy např. v Indii, Rusku a Wolfsburgu je použit stejný standard pro jejich výstavbu. Tímto je přidán do oblasti údržby jednotný stav a minimalizují se tím náklady na sklad či náhradních komponent. Další výhodou je řízení skrz řídicí panel, který je řízen na stejné bázi v celém koncernu, což přináší lepší orientaci a flexibilitu, např. pro údržbáře, kteří se přesouvají mezi výrobními podniky.

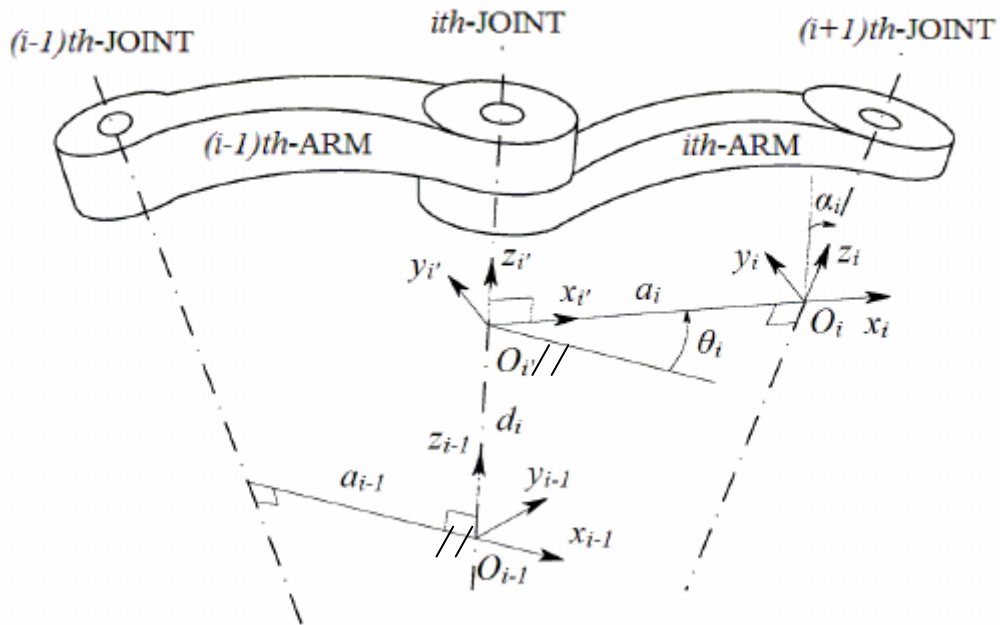
V rámci oblasti svařovny je nejčastěji používána základní sada technologií, která obsahuje např. odporové svařování (především bodové svařování), tucker (navařený čep, který slouží k upevnění dalších dílů), svařování laserem či lepení. Tato sada je nutná k montáži a svaření karoserie. Používání neustále stejné technologie umožňuje vytvoření a předpřípravení standardu, který je pak možné aplikovat plošně. Vývoj a testování standardizovaných prvků je prováděn v Německu. Pro oblast standardizace je vymezeno specializované centrální oddělení, resp. laboratoře, kde jsou jednotlivé komponenty testovány a vyvíjeny. Oddělení plánování svařovny každý rok dostává podklady, tj. aktualizované standardy, podle kterých se řídí a provádí požadované změny ve svařovně. Velká výhoda je ušetření financí v rámci závodu. Příklady standardizací, např. topologie zasítování, viz příloha [C.1] nebo standard pro formát označování profinetu, viz příloha [C.2].

3 Kinematika robota a Denavit-Hartenbergovy metody

Obecně kinematika je část mechaniky, která se zabývá pohybem těles, bez ohledu na síly nebo momenty, které pohyb způsobují. Pohyb může být jakákoliv změna polohy nebo orientace tělesa. Pokud je známá poloha v čase, tak lze pomocí derivace získat rychlost a zrychlení. Zde bude popsána především kinematika robotů, která se zabývá geometrií pohybu robota a trajektorie, po kterých se pohybují jednotlivé body, tj. především sledování polohy, popř. rychlosti. Pro popis kinematiky robotů existuje několik různých souřadnicových systémů. Nejpoužívanější souřadnicový systém pro robotický systém je kartézská soustava souřadnic, ale existují i sférické nebo cylindrické. Pomocí těchto systémů jsou definovány body v pracovním prostoru. Je nutno rozlišovat mezi absolutními prostorovými souřadnicemi, souřadnicemi stroje a chapadla. Polohy bodů v pracovním prostoru jsou stanoveny pomocí pravotočivého souřadnicového systému a vztahují se vždy k počátku souřadnicového systému (globálně), tedy k základně robota. V případě robotů s kloubovými rameny se určí poloha os pomocí úhlu natočení ramen (α , β , γ). Jedná se o souřadnice, které se vztahují k robotu. Počátek souřadného systému vztaženého k chapadlu je vždy ve středu chapadla, viz kapitola 3.2. Obecně lze říci, že každé části robota je přiřazena jedna či více lokálních soustav, které umožní popis robotického systému. Nyní lze zavést pojem tzv. transformační vztah, který vyjadřuje přepočítání mezi jednotlivými souřadnými systémy a o kterém bude řeč v následující části.

Transformace mezi dvěma kartézskými souřadnými systémy lze rozdělit na rotaci a translaci (posun). Rotace může být reprezentována mimo jiné např. pomocí Eulerových úhlů, Gibbsova vektoru, ortonormálních matic a Hamiltonových quaternionů. Nejvíce používané jsou homogenní transformace, založené na maticích 4×4 (ortonormální matice). Denavit a Hartenberg dokázali, že obecná transformace mezi dvěma klouby vyžaduje pouze čtyři parametry. Tyto parametry, nyní známé jako DH parametry se staly základním nástrojem pro robotickou analýzu. Pomocí DH lze přecházet mezi dvěma souřadnými systémy pomocí rotace osy X (R_x), posunu v jejím směru (P_x), rotace podél osy Z (R_z) a posunu v jejím směru (P_z).[9][14][16]

3.1 Obecný algoritmus a odvození DH metody



Obrázek 7: Kinematika DH [11]

Z obrázku kinematiky DH (viz Obrázek 7) bude dále popsán algoritmus DH metody:

- Nalezneme osy kinematických dvojic a jejich společné normály.
- Označíme osu i jako z_{i-1} a osu $i + 1$ jako z_i .
- Vyznačíme počátek souřadnic O_i v průsečíku osy z_i s normálou z_i a z_{i-1} .
- Podobně vyznačíme počátek souřadnic O_i' a O_{i-1} v ose z_i' a z_{i-1} .
- Vyznačíme vzdálenosti a_{i-1}, d_i, a_i .
- Osu x_i volíme tak, aby směřovala ve směru normály z kloubu i do kloubu $i + 1$.
- Osu y_i určíme tak, aby byl souřadný systém pravotočivý.
- Provedeme rotaci kolem osy z_{i-1} o úhel θ_i tak, aby osy x_{i-1} a x_i měli stejnou orientaci.
- Provedeme translaci podél osy z_{i-1} o d_i , tak aby se osy x_{i-1} a x_i' překryly.
- Provedeme translaci podél osy x_i o a_i do bodu O_i .
- Provedeme rotaci podél osy x_i o úhel α_i , tak aby se osy z_{i-1} a z_i překryly.

Nyní lze definovat tabulku DH parametrů s informacemi, zdali je daný parametr proměnná či konstanta v závislosti na rotaci a posunu (viz Tabulka 4). Úhel kloubu θ_i je určen natočením osy x_i okolo osy z_{i-1} , tak aby osa x_{i-1} byla \parallel s osou x_i . Vzdálenost kloubu d_i je určena posunutím podél osy z_{i-1} , tak aby se osa x_{i-1} protнула s osou x_i .

Délka ramene a_i je vytvořena posunutím podél osy x_i , potřebným k dosažení osy z_{i-1} , tak aby se protнула s osou z_i . Twistový úhel ramene α_i je dán natočením okolo osy x_i , potřebným k tomu, aby osa z_{i-1} byla \parallel s osou z_i .

Tabulka 4: Tabulka DH parametrů

Parametr	Označení	Rotace	Translace
úhel kloubu	θ_i	prom.	konst.
vzdálenost kloubu	d_i	konst.	prom.
délka článku	a_i	konst.	konst.
twistový úhel	α_i	konst.	konst.

Následně je možno přes odvozené matice rotace a posunu [12] snadno určit obecnou transformační matici pro jedno rameno: ${}^{i-1}_i T = R_Z(\theta_i) \cdot T_Z(d_i) \cdot R_x(\alpha_i) \cdot T_x(a_i)$, tj.

$${}^{i-1}_i T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Vynásobením všech matic získáme požadovaný DH transformační převod, tj.

$${}^{i-1}_i T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

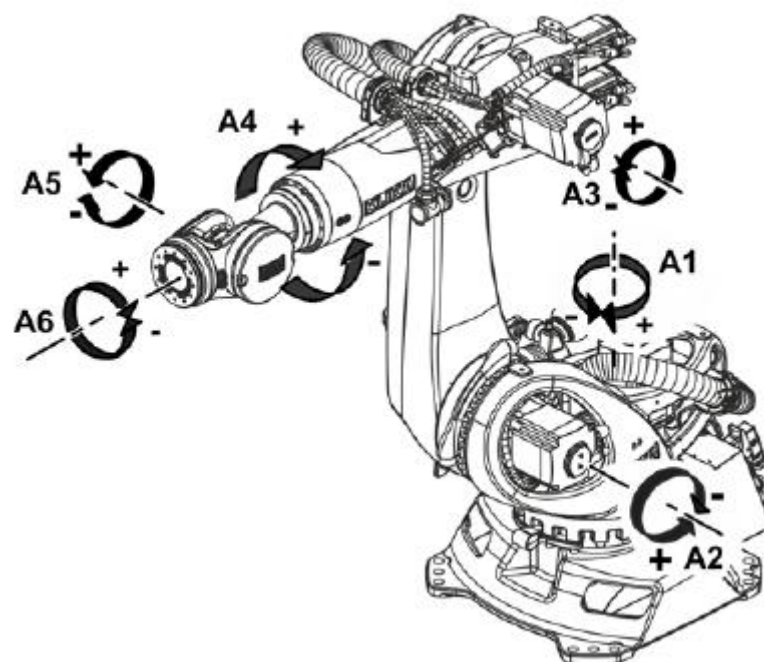
3.2 Přímá úloha kinematiky ve spojení s DH metodou

Přímá úloha kinematiky využívá kinematických rovnic pro výpočet polohy koncového členu robota (chapadla) v závislosti na poloze a úhlech natočení jednotlivých členů. Označíme-li vektor kloubových souřadnic $q = (q_1, q_2, q_3, q_4, q_5, q_6)$ a vektor koncového členu robota, např. chapadla $G = (x, y, z, a, b, c)$, pak existuje zobrazení z prostoru kloubových souřadnic do prostoru poloh chapadla (kartézských souřadnic). Matematicky jej lze vyjádřit následujícím vztahem:

$$G = f(q)$$

Tento vztah představuje 6 rovnic, které je třeba sestavit. Nevznikají zde žádné problémy při odvozování rovnic, proto vždy existuje řešení přímé kinematické úlohy pro manipulator právě nalezením těchto rovnic.

V této práci je využit 6 osý robot, obecné schéma (viz *Obrázek 8*). Kinematická struktura robota je angulární – má 6 rotačních os, 6 vzájemně otočených a posunutých souřadných systémů. Tento typ robota je v praxi nepoužívanější a mimo výše uvedené výhody v kapitole 1 robot disponuje např. nižší cenou při výrobě rotačních částí, vysokou rychlostí manipulačních pohybů, minimální zastavěnou plochou či snadným rozšířením pracovního prostoru (pojezd). [12]



Obrázek 8: Angulární kinematická struktura robota

Jak již bylo řečeno výše, problém přímé úlohy kinematiky se týká vztahu mezi jednotlivými klouby manipulatoru, polohou (pozicí) a orientací nástroje. Sériový manipulator je složen ze základny (base), která je napevno spojena se zemí, dále z několika ramen, které jsou spojeny klouby a koncového efektoru (chapadla). Tímto spojením je zajištěn řízený či volný pohyb dvou ramen, které kloub spojuje. Rameno je pevná část robota, která definuje prostorový vztah mezi dvěma sousedními společnými osami. Podrobný popis manipulatoru viz 2.3. Cílem této úlohy je určení výsledné polohy koncového členu ze všech kloubových proměnných kinematického řetězce s pomocí odvozené DH transformace v kapitole 3.1. Přímá úloha kinematiky je řešena vzájemným vynásobením

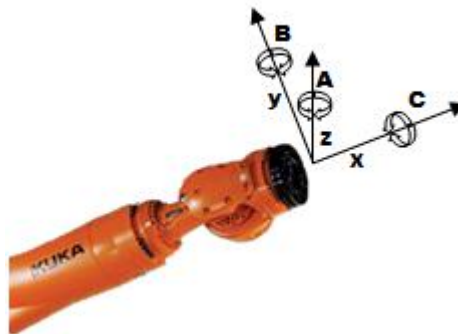
všech transformačních matic ${}^{i-1}_iT$ v řetězci mezi základnou a koncovým efektořem (chapadlem), tj.

$$\text{základna}_{\text{efektor}}T = {}^0_1T \cdot {}^1_2T \cdot {}^2_3T \dots {}^{n-1}_nT$$

V případě 6 osého robota platí, že polohu a orientaci jeho efektořu vzhledem k souřadnicovému systému základny, lze zapsat následovně:

${}^0_6T(q) = {}^0_1T(q_1) \cdot {}^1_2T(q_2) \cdot {}^2_3T(q_3) \cdot {}^3_4T(q_4) \cdot {}^4_5T(q_5) \cdot {}^5_6T(q_6)$, kde q_i je kloubová souřadnice posuvného nebo otočného kloubu, pro kloub $i = 1$ až 6.

Důležitá je koncová poloha chapadla, kterou je potřeba hledat (viz *Obrázek 9*). Tato poloha má 6 stupňů volnosti, tedy 6 je minimální počet nezávislých parametrů pro jednoznačný popis systému. Osy x, y, z značí kartézský souřadnicový systém chapadla, který je vztažen k počátku souřadnicového systému (základně robota). Označení A, B, C odpovídá rotaci kolem příslušné osy.[14][15]



Obrázek 9: Kartézské souřadnice chapadla

3.3 Implementace přímé úlohy robota

Řešení přímé úlohy je implementováno v jazyce C# a bude součástí kompletního aplikačního SW. Při řešení je využít výpočet Eulerových úhlů ψ , θ a ϕ z matic rotace $R_x(\psi)$ okolo os x , $R_y(\theta)$ okolo osy y a $R_z(\phi)$ okolo osy z .

Matice R_x a R_z jsou již uvedeny pro výpočet transformační matice jednoho ramene v kapitole 3.1. Zbývá definovat matici R_y okolo osy y :

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Obecnou matici rotace R lze zapsat v této formě:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

Dále je potřeba získat matici $R = R_z(\phi) \cdot R_y(\theta) \cdot R_x(\psi)$, tj.

$$R = \begin{bmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix}$$

Nyní lze určit hodnoty Eulerových úhlů s pomocí postupu uvedeného v literatuře [17].

Obecně lze postupovat těmito kroky:

- Nalezneme úhly pro θ , tak že položíme $R_{31} = -\sin\theta$, tedy $\theta = -\sin^{-1}(R_{31})$
- Dále je nutno určit hodnoty úhlů θ_1, θ_2 pro $R_{31} = \pm 1$, tzn. $\theta_1 = -\sin^{-1}(R_{31})$ a $\theta_2 = \pi - \theta_1 = \pi + \sin^{-1}(R_{31})$.
- Nalezneme úhly ψ , tak že $\frac{R_{32}}{R_{33}} = \tan(\psi)$, pak $\psi = \text{atan2}(R_{32}, R_{33})$.

Funkce $\text{atan2}(y, x)$ vrací navíc to, že znaménka obou argumentů jsou použita k vypočtení příslušného kvadrantu výsledku, který leží v intervalu $(-\pi, \pi]$.

Pro nalezení hodnot úhlů ψ_1, ψ_2 využijeme rovnici $\psi = \text{atan2}\left(\frac{R_{32}}{\cos\theta}, \frac{R_{33}}{\cos\theta}\right)$, platí pouze v případech, kdy $\cos\theta \neq 0$. Pak $\psi_1 = \text{atan2}\left(\frac{R_{32}}{\cos\theta_1}, \frac{R_{33}}{\cos\theta_1}\right)$ a $\psi_2 = \text{atan2}\left(\frac{R_{32}}{\cos\theta_2}, \frac{R_{33}}{\cos\theta_2}\right)$.

- Již zbývá nalézt úhly ϕ , tak že $\frac{R_{21}}{R_{11}} = \tan(\phi)$, pak $\phi = \text{atan2}\left(\frac{R_{21}}{\cos\theta}, \frac{R_{11}}{\cos\theta}\right)$, opět platí pouze v případech, kdy $\cos\theta \neq 0$. Pak $\phi_1 = \text{atan2}\left(\frac{R_{21}}{\cos\theta_1}, \frac{R_{11}}{\cos\theta_1}\right)$ a $\phi_2 = \text{atan2}\left(\frac{R_{21}}{\cos\theta_2}, \frac{R_{11}}{\cos\theta_2}\right)$.
- Pro všechny případy, kdy $\cos\theta \neq 0$ dostáváme dvě řešení: $(\psi_1, \theta_1, \phi_1)$ a $(\psi_2, \theta_2, \phi_2)$.

Tento postup nefunguje v případě, že prvek R_{31} z matice rotace je roven 1 nebo -1, což odpovídá při $\theta = \pm \frac{\pi}{2}$. Pokud by se tento princip použil pro hledání úhlů ψ a ϕ , tak prvky R_{11}, R_{21}, R_{32} a $R_{33} = 0$. V tomto případě ($\cos\theta = 0$) prvky neomezují hodnoty úhlů ψ a ϕ . Z toho důvodu je nutné použít jiné prvky z matice rotace pro výpočet úhlů ψ a ϕ .

- f) Předpokládejme, že $\theta = \frac{\pi}{2}$. Sestavíme rovnice, kde $R_{12} = \sin(\psi - \phi)$ a $R_{13} = \cos(\psi - \phi)$. Jejich úpravou dostaneme následující vztah:

$$\psi = \phi + \operatorname{atan2}(R_{12}, R_{13})$$

- g) Druhá možnost je, že $\theta = -\frac{\pi}{2}$, zde použijeme stejný postup jako v předchozím bodě. Po úpravě dostaneme vztah:

$$\psi = -\phi + \operatorname{atan2}(-R_{12}, -R_{13})$$

Ukázka implementace v jazyce C# (viz *Zdrojový kód 1*).

```
if (U[2, 0] != 1 && U[2, 0] != -1)//vypocet eulerovych uhlu z matice rotace
{
    theta1R = -Math.Asin(U[2, 0]);
    theta2R = Math.PI - theta1R;
    psi1R = Math.Atan2(U[2, 1] / Math.Cos(theta1R), U[2, 2] / Math.Cos(theta1R));
    psi2R = Math.Atan2(U[2, 1] / Math.Cos(theta2R), U[2, 2] / Math.Cos(theta2R));
    fi1R = Math.Atan2(U[1, 0] / Math.Cos(theta1R), U[0, 0] / Math.Cos(theta1R));
    fi2R = Math.Atan2(U[1, 0] / Math.Cos(theta2R), U[0, 0] / Math.Cos(theta2R));
}
else
{
    fiR = 0;
    if (U[2, 0] == -1)
    {
        thetaR = Math.PI / 2;
        psiR = fiR + Math.Atan2(U[0, 1], U[0, 2]);
    }
    else
    {
        thetaR = -Math.PI / 2;
        psiR = -fiR + Math.Atan2(-U[0, 1], -U[0, 2]);
    }
}
```

Zdrojový kód 1: Ukázka kódu pro výpočet Eulerových úhlů z matice rotace

4 Vývoj SW pro sběr technologických dat z robota

Aplikace byla vyvíjena v jazyce C# v prostředí Microsoft Visual Studio 2017. C# je považován za nejmodernější jazyk. Microsoft dodává kompletní sadu knihoven, ve kterých jsou předpřipraveny řady struktur a komponent, např. pro práci s konzolí, formuláři, prvky, databázemi apod. Další nespornou výhodou je, že Microsoft je tvůrcem Windows. Jejich komponenty jsou pro tento systém tedy velmi dobře odladěné.

Aplikace komunikuje na bázi TCP, resp. metodě připojení klient/server. Zjednodušeně to znamená, že je vytvořen server, který bude naslouchat nějakému portu. K tomu je vytvořen klient, který bude data odesílat na příslušný port, kterému server naslouchá. TCP vytváří virtuální okruh mezi koncovými aplikacemi, tedy spolehlivý přenos dat. Je to plně duplexní spojení, tj. je zajištěn současný obousměrný přenos dat.

Připojení bylo realizováno pomocí ethernetového (modrého) UTP kabelu, zapojeného do síťového přepínače, napojeného na sběrnici, resp. profinet (viz Obrázek 10). Profinet je průmyslová komunikační sběrnice pro řídicí systémy v oblasti průmyslové automatizace.



Obrázek 10: Připojení kabelu do sběrnice

4.1 Struktura programu

Program je rozdělen do 5 tříd pro lepší přehlednost. V hlavní třídě *Form1.cs* jsou zapsány potřebné metody, funkce a události, které jsou propojeny s jednotlivými komponenty formuláře ve *Form1.Designer.cs* a určují tak jejich chování. Třída *KVP.cs* zajišťuje komunikaci a metodu na čtení dat z robota. Třída *ClassDH.cs* řeší přímou úlohu robota, viz kapitola 3.3. Následující třída *Form2.cs* je určena pro výpis DH matice. Poslední

třída *Graph.cs* obsahuje metodu pro vytvoření grafu, který reprezentuje vybrané instrukce a zobrazuje je v grafu. Podrobný popis významných metod je uveden v následujících kapitolách.

4.1.1 Realizace navázání komunikace TCP

V jazyce C# je potřeba vytvořit direktivu na třídu *System.Net.Sockets*, která zajišťuje použití *TcpClient*, tedy jeho klientské připojení pro síťové služby protokolu TCP. Tato třída je nezbytná pro připojení ke vzdálenému hostiteli. Implementace do jazyka C# (viz *Zdrojový kód 2*). Zahájení asynchronního požadavku pro připojení vzdáleného hostitele zajišťuje metoda *KVPSSocket.BeginConnect*. Vlastnost *AsyncWaitHandle.WaitOne* čeká na dokončení asynchronní operace s daným časovým rámcem. Pokud je spuštěn *KukavarProxy*, spojení trvá v řádech milisekund. Pokud *KukavarProxy* spuštěn není nebo je ethernetový kabel odpojen, pak *KVPSSocket.Connected* vrátí hodnotu *FALSE* a uživateli je zobrazena informační hláška. Podrobnosti o *KukavarProxy* jsou uvedeny v kapitole 4.1.2.

Součástí programu je také metoda *Disconnect*, která ukončí připojení vzdáleného hostitele a uvolní veškeré prostředky, které jsou přidružené ke koncovému bodu připojení, tzv. *Socketu*.

```
public TcpClient KVPSSocket = new TcpClient();

public bool Connect(string IPAddress, int Port, int timeout)
{
    try
    {
        KVPSSocket.SendTimeout = timeout;
        KVPSSocket.ReceiveTimeout = timeout;
        // Connect to server using a timeout
        IAsyncResult result = KVPSSocket.BeginConnect(IPAddress, Port, null, null); //připojení k hostiteli
        result.AsyncWaitHandle.WaitOne(timeout, true);
    }
    catch (Exception)
    {
        return false;
    }

    return KVPSSocket.Connected;
}
```

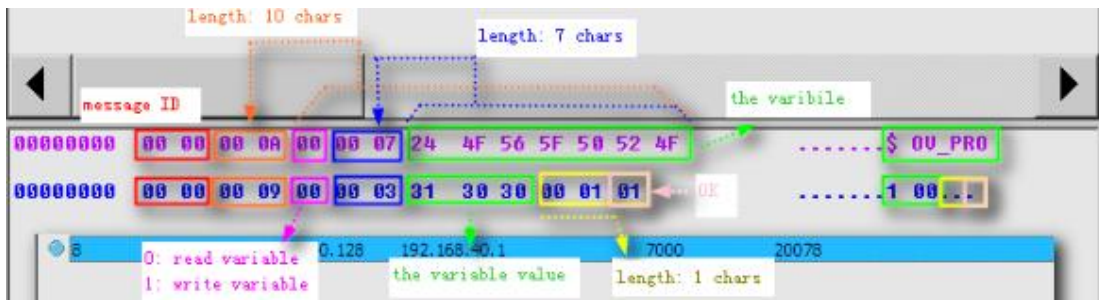
Zdrojový kód 2: Metoda pro připojení k hostiteli

4.1.2 Metoda pro čtení dat z robota

Čtení dat probíhá na základě vyslané instrukce, kterou lze obecně vyjádřit ve tvaru „*\$X_Y*“. Kompletní nepublikovaný seznam instrukcí je uveden v literatuře [18].

Metoda, která zajišťuje čtení údajů, je nazývána *readVar*. Samotné čtení znaků je

naprogramováno dle předepsaného formátu (viz *Obrázek 11*). Zpráva je rozdělena do několika sekvencí, které jsou zasílány na výstup.



Obrázek 11: Formát pro přenos dat [20]

Nejprve je zvolená instrukce zakódována do sekvence bytů a uloží se do pole. Dále jsou vytvořena bytová pole o velikostech messageID, celkové délce zprávy, funkce a délce žádané hodnoty. Poté jsou přiřazeny příslušné velikosti do bytového pole a je vytvořeno nové pole *msg*, které bude obsahovat všechny informace dle výše uvedeného formátu. Pomocí metody *KVPSSocket.GetStream()* lze odesílat a přijímat data, resp. jsou zapsány sekvence bytů do aktuálního datového proudu. Dále je přečten celkový počet bytů do vyrovnávací paměti z datového proudu a je nastaven jako velikost výsledného pole *result*. Poté je procházeno bytové pole a dochází k separaci žádané hodnoty od ostatních, která je uložena do pole *result*. Metoda *readVar* nakonec dekóduje všechny byty z výsledného pole do řetězce (stringu) (viz *Zdrojový kód 3*).

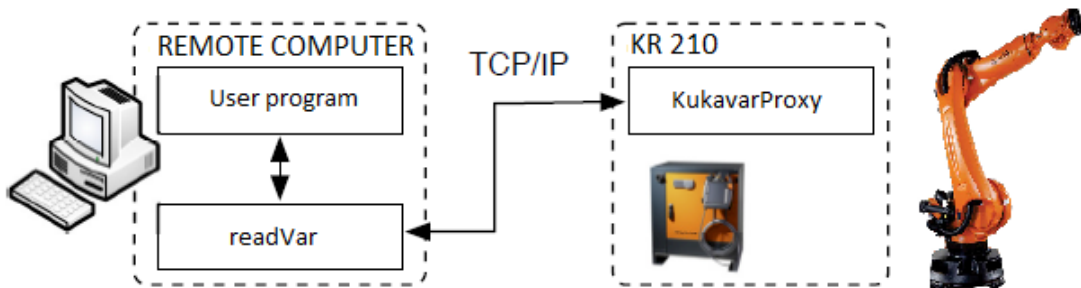
```

public string readVar(string name)
{
    string q = name;
    byte[] data = Encoding.Default.GetBytes(q);
    byte[] id = new byte[2];
    byte[] lengthAll = new byte[2];
    byte[] fce = new byte[1];
    byte[] lengthData = new byte[2];
    int l = data.Length;
    int ll = l + 3;
    lengthData[1] = (byte)l;
    lengthAll[1] = (byte)ll;
    fce[0] = (byte)0;
    byte[] msg = id.Concat(lengthAll).Concat(fce).Concat(lengthData).Concat(data).ToArray();
    NetworkStream stream = KVPSSocket.GetStream();
    stream.Write(msg, 0, msg.Length);
    byte[] buffer = new byte[255];
    int iRx = stream.Read(buffer, 0, buffer.Length);
    byte[] result = new byte[iRx];
    for (int i = 7; i < iRx; i++)
    {
        result[i - 7] = buffer[i];
    }
    return Encoding.Default.GetString(result);
}

```

Zdrojový kód 3: Metoda readVar

Pro výměnu dat mezi uživatelem a robotem existuje open-source software JOpenShowVar, který je napsán v jazyce Java. Na straně firemních PC ve Škoda Auto však výměna dat pomocí tohoto SW nefungovala. Byla tedy vyvinuta aplikace s vlastní metodou *readVar* v jazyce C#. K uskutečnění celého spojení je nutno mít spuštěný již zmiňovaný KukavarProxy. Tento software lze na robotech spouštět pouze s licencí od výrobce KUKA. Řídicí systém robota je chráněn proti neznámému či škodlivému SW pomocí metody „whitelistingu“ KUKA.CPC. KukavarProxy je možno definovat jako server neboli bránu pro komunikaci mezi ethernetem a TCP s roboty KUKA, který umožňuje každému klientovi TCP zobrazit (číst) a popř. měnit hodnoty proměnných robota. Jednoduché blokové schéma mezi KukavarProxy a PC (viz *Obrázek 12*).



Obrázek 12: Blokové schéma komunikace

4.1.3 Metoda pro ukládání do souboru

Ukládání do souboru je řešeno v hlavní třídě *Form1.cs*, v metodě *Save_Click*. Po stisknutí tlačítka *Save* je následně otevřeno dialogové okno pro možnost zadání názvu, typu a umístění souboru. Lze si vybrat mezi formáty typu CSV a TXT. CSV formát je navržen pro širokou podporu mezi operačními systémy a aplikacemi. Je především určen pro ukládání tabulkových informací do souboru s odděleným textem. Menší nevýhodou je, že neumožňuje formátování textu, např. oproti souborům typu XLS. Rozhodující výhodou je, že soubory typu CSV mohou být otevírány a editovány v textových editorech, zatímco soubory uložené v aplikaci excel nikoliv. Pro lepší přehlednost a orientaci v tabulce je doporučeno ukládat soubor do CSV a následně zobrazit v excelu nebo klasicky do textového formátu TXT pro následné zpracování.

V metodě je definována převodní tabulka pro přiřazení jednotek k měřené veličině. Část kódu pro zápis hodnot do souboru (viz *Zdrojový kód 4*).

```

Dictionary<string, string> UnitConvertTable = this.UnitConvertTable;
if (CurrentSampleCount == 1)
{
    sw.Write("t [ms];");//hlavička času
    for (int i = 0; i < SelectedInstructions.Items.Count; i++)//hlavička tabulky
    {
        string Instrukce = SelectedInstructions.Items[i].ToString();
        sw.Write(Instrukce);
        if (UnitConvertTable.ContainsKey(Instrukce))
        {
            sw.Write "[" + UnitConvertTable[Instrukce] + "];";
        }
        sw.Write(";");
    }
    sw.WriteLine();
}
}

```

Zdrojový kód 4: Část kódu pro zápis dat do souboru

Do prvního sloupce uloženého souboru je vždy vypsán aktuální čas naměřené veličiny. Další sloupce obsahují zvolené instrukce (veličiny), které si uživatel zvolí v libovolném pořadí v aplikaci. Na konec souboru je přiřazen údaj o celkové době měření a počtu vzorků (viz Obrázek 13).

	A	B	C	D	E
1	t [ms]	\$POS_ACT	\$POS_ACT	\$IN[1]	\$CURR_ACT[3]
2	1000	822,667	-4,455	FALSE	0,00801
3	2000	822,667	-4,455	FALSE	0,00744
4	3000	822,667	-4,455	FALSE	0,00648
5	4000	822,667	-4,455	FALSE	0,00782
6	5000	822,667	-4,455	FALSE	0,00858
7	6000	822,667	-4,455	FALSE	0,0082
8	7000	822,667	-4,455	FALSE	0,00839
9	8000	822,667	-4,455	FALSE	0,00877
10	9000	822,667	-4,455	FALSE	0,00839
11	10000	822,667	-4,455	FALSE	0,00782
12	The time of measurement: 10 s				
13	The number of samples: 10				

Obrázek 13: Struktura souboru

4.1.4 Metoda pro nastavení vzorkovací periody

Metoda *TimerMereni_Tick* slouží pro nastavení vybrané vzorkovací periody T při zvoleném celkovém čase t . Část implementace v jazyce C# (viz Zdrojový kód 5). Pro eliminaci zpoždění je mezi jednotlivými vzorky zvolen následující algoritmus:

1. Zjistí celkový čas t a periodu T .
2. Určí cílový počet vzorků, vztahem: $\frac{t}{T}$.
3. Zjistí čas t_{exp} při aktuálním vzorku.
4. Zjistí skutečný uplynulý čas t_{curr} od startu.
5. Určí zpoždění $t_{del} = t_{curr} - t_{exp}$.

6. Nastav interval (periodu) měření, tak že $T - t_{del}$.
7. Pokud je aktuální vzorek roven cílovému počtu vzorků, tedy byla zapsána poslední hodnota, zastav měření.

```
private void TimerMereni_Tick(object sender, EventArgs e)
{
    if(!kvp.KVPSocket.Connected)
    {
        TimerMereni.Enabled = false;
        return;
    }

    double SelectedTime = double.Parse(Times.SelectedItem.ToString());
    double SamplePeriod = double.Parse(Period.SelectedItem.ToString()) / 1000.0;
    int TargetCount = (int)Math.Round(SelectedTime / SamplePeriod);
    bool Konec = false;
    try
    {
        StreamWriter sw = new StreamWriter(SelectedOutputFile, true);
        double ValueForGraph;
        int WantedPeriod = int.Parse(Period.SelectedItem.ToString());
        int ExpectedTime = WantedPeriod * CurrentSampleCount;
        int CurrentTime = Environment.TickCount - StartTime;
        int Delay = CurrentTime - ExpectedTime;
        TimerMereni.Interval = Math.Max(1, WantedPeriod - Delay);
    }
}
```

Zdrojový kód 5: Metoda Timer

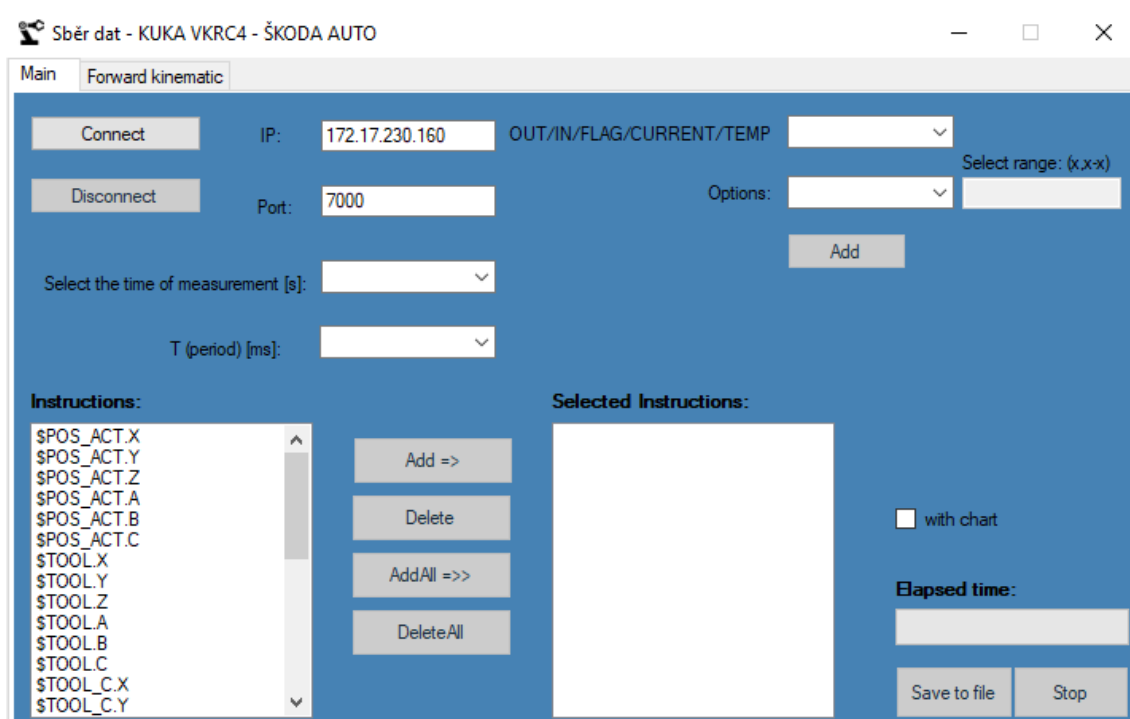
Doba získání jednoho vzorku (včetně zápisu do souboru) se pohybovala v rozmezí od 4 do 15 ms. V určitých fázích měření se doba zvýšila až na hranici 90 ms. To může být způsobeno kolizemi na sběrnici, resp. profinetu či na straně robota, který je v daném okamžiku zaneprázdněn a není schopen odpovídat v dostatečném limitu. Klíčovým parametrem zpoždění je počet měřených veličin, dále pak počet připojených účastníků na sběrnici nebo vytížení RAM paměti. Bylo provedeno několik měření s různými vstupními parametry (doba měření, vzorkovací perioda). Z těchto měření vyplynulo, že optimální perioda pro např. 12 měřených veličin je v rozsahu 200 – 300 ms. Je nutno volit vhodnou vzorkovací periodu v závislosti na počtu zvolených měřených veličin. Vzhledem k nahodilým pádům komunikace robota (v automatickém režimu při periodách do 200 ms), bylo nutno zvýšit periodu na 300 ms. Během testování při této periodě k výpadkům komunikace již nedocházelo. V naprosté většině variant měření bylo zpoždění při této hranici zanedbatelné. Pro účely aplikace, tj. měření polohy a proudů je tato hranice dostatečná.

4.2 Popis aplikace

Aplikace je rozdělena do dvou oken, resp. záložek *Main* a *Forward kinematic*. Uživatel si může libovolně přepínat mezi těmito okny dle vlastní potřeby a účelu měření.

4.2.1 Hlavní okno „Main“

Hlavní okno aplikace je navrženo jednoduše a přehledně (viz *Obrázek 14*). Uživatel si nejprve zvolí IP adresu robota, na kterou se chce připojit. Pro úspěšné připojení a přenos dat je potřeba vždy zadat správné číslo portu. Hodnoty v aplikaci jsou již přednastaveny pro konkrétního robota KUKA, umístěného v trénovací buňce ve svařovně.

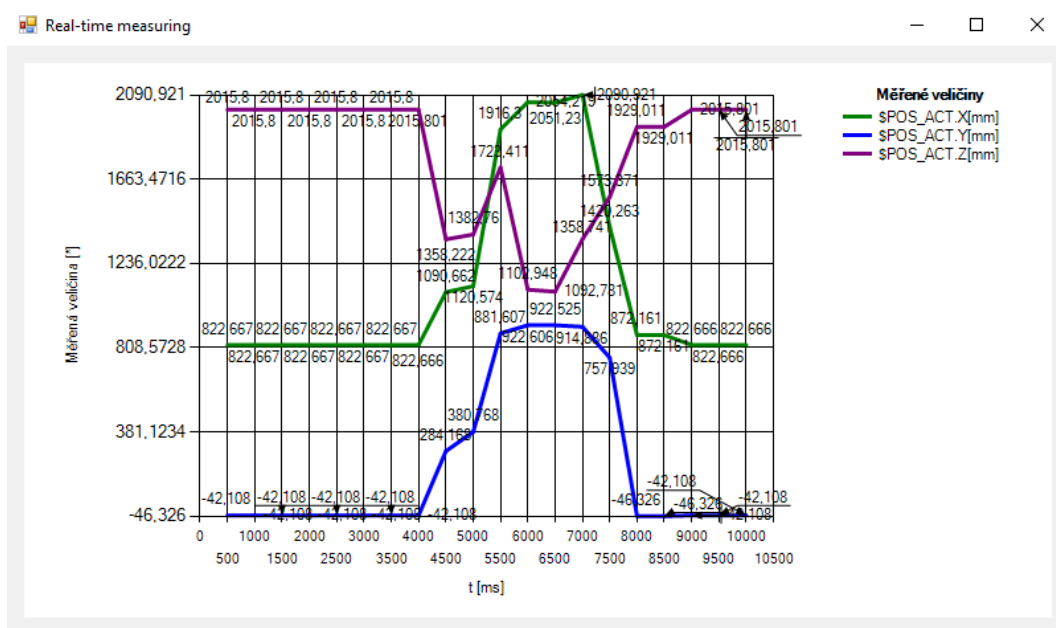


Obrázek 14: Hlavní okno aplikace

Je nutné, aby byla na robota zároveň spuštěna aplikace KukavarProxy. V případě správného připojení je tlačítko *Connect* přepnuto do zelené barvy a současně se aktivují ostatní tlačítka ve formulářovém okně. Uživatel si následně označí v levém okně (listboxu) požadované instrukce, které definují měřenou veličinu a pomocí tlačítka *Add* si je přesune do okna vybraných instrukcí. V případě, že by chtěl přesunout všechny instrukce najednou, slouží k tomu tlačítko *AddAll*. Z pravého okna lze označené instrukce vymazat tlačítkem *Delete* nebo kompletně celé okno, tlačítkem *DeleteAll*. Seznam obsahuje instrukce pro měření polohy robota, popř. nástroje v kartézských souřadnicích (x, y, z), úhlu otočení kolem příslušných os nástroje (a, b, c) a úhlů θ pro zjištění aktuální pozice jednotlivých os. Dále je možno měřit aktuální dráhovou rychlost, startovací a cílovou pozici aktuálního pohybového bloku. Teoreticky lze měřit jakoukoliv hodnotu ze seznamu [18]. V pravé horní části jsou umístěny instrukce, které vyžadují výběrové pole. Instrukci vstupů IN a OUT je možno navolit v rozsahu 1 až 4096, instrukci flagů v rozsahu 1 až 1096. Instrukce CURRENT v rozsahu 1 až 6, reprezentuje proud ve

všech 6 osách robota. Přepočítání na aktuální proud jednotlivých os v cíleném formátu, viz příloha [B]. Poslední instrukce TEMP měří teplotu servomotorů jednotlivých os. Po výběru aktuální instrukce z horní nabídky je nutno přidat hodnotu do seznamu zvolených instrukcí opět pomocí tlačítka *Add*, umístěného v horní části okna. Pro rychlé navolení vyššího počtu hodnot je k dispozici pole pro zvolení rozsahu, popř. konkrétní hodnoty.

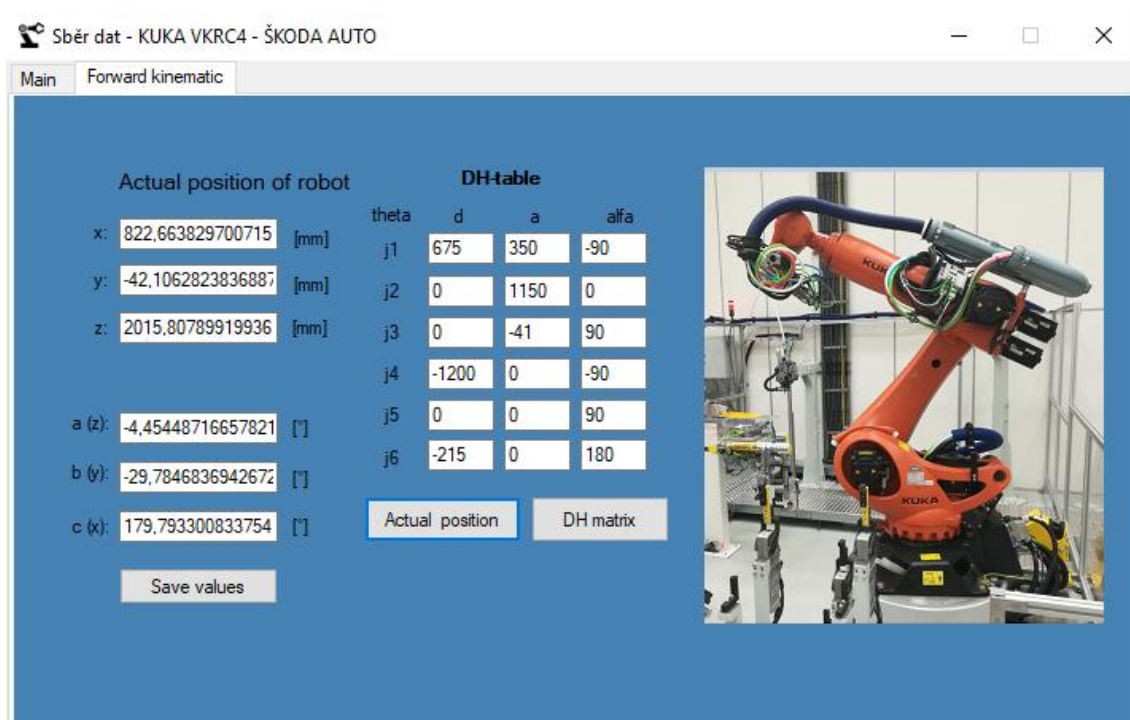
V další části si uživatel zvolí čas celkového měření a to v různých intervalech v rozsahu 10 – 60 s. Poté si nastaví vzorkovací periodu, resp. interval získávaných vzorů v rozsahu 100 – 2000 ms. Pokud uživatel zapomene vyplnit některé pole, je zobrazeno okno s informační hláškou. Aktuální čas měření je zobrazen za popiskem *Elapsed Time* a k jeho vizualizaci slouží i *progressbar*, který je umístěn hned pod ním. Po kliknutí na tlačítko *Save to File* je zobrazeno dialogové okno, kde si uživatel vyplní název nově vytvořeného souboru a jeho typ. Po potvrzení je spuštěno měření v požadované délce, jakou si uživatel zvolil. Po uplynutí zvolené doby měření je soubor uložen na požadované místo a je zobrazena informační hláška. V případě, že uživatel zaškrtnul před spuštěním měření *checkbox* s názvem „with chart“, tak je na konci měření zobrazeno okno s grafem, které reprezentuje naměřené hodnoty uložené v souboru (viz *Obrázek 15*). Okno s grafem obsahuje v pravé části zvolené veličiny (instrukce) v různých barvách, konkrétně polohu v kartézských souřadnicích. Okno umožňuje vyříznutí libovolného průběhu měření a tím zobrazuje přehlednější průběh křivky. Na ose *x* se nachází čas *t* v ms. Osa *y* odkazuje na měřenou veličinu včetně jednotek z pravé horní části.



Obrázek 15: Ukázka grafu

4.2.2 Okno Forward kinematic

Okno (viz *Obrázek 16*) je navrženo pro výpočet přímé úlohy z DH tabulky. DH tabulka obsahuje přednastavené hodnoty pro robota KR210 R2700 EXTRA C4 FLR. Každý typ robota KUKA má ve svém adresáři (R1\Mada\\$machine.dat) uložen soubor s parametry DH tabulky, podle které uživatel zadá hodnoty do aplikace a následně vypočte aktuální pozici robota, která je zobrazena v levé části okna (textboxů). Pro zajímavost je možno zobrazit okno s výslednou DH maticí. Hodnoty je možno uložit opět do souboru pomocí tlačítka *Save values* (viz *Obrázek 17*).



Obrázek 16: Okno Forward kinematic

Výpočet lze ověřit s naměřenými hodnotami během klasického měření. Výhodou je zjištění hodnot aktuální pozice a úhlu otočení kolem robota bez nutnosti spouštět celé měření. Výpočet pomocí Eulerových úhlu se oproti skutečné hodnotě lišil v řádu jednotek tisícín.

	A	B	C	D
1	Vypočtena pozice pomocí Eulerových uhlu			
2	x [mm]:	822,6638		
3	y [mm]:	-42,1063		
4	z [mm]:	2015,808		
5	a (z) [°]:	-4,45449		
6	b (y) [°]:	-29,7847		
7	c (x) [°]:	179,7933		

Obrázek 17: Ukázka vypočtených hodnot

5 Způsoby využití SW pro optimalizaci robotických aplikací

Software byl vyvinut primárně pro diagnostiku chybových stavů mezi průmyslovým robotem KUKA a technologií v podniku Škoda Auto v Mladé Boleslavi. Místem pracoviště byla trénovací buňka ve svařovně (hala M14). Buňka obsahuje dva trénovací roboty, tj. robota KUKA a robota FANUC (viz *Obrázek 18*).



Obrázek 18: Trénovací buňka

Robotické svařovací jednotky, resp. svařovací kleště jsou používány ve vysoce automatizované výrobě automobilového průmyslu. Dle přísných požadavků na kvalitu při svařování a pro zajištění 100% spolehlivosti jednotlivých komponent a svařovacích robotů je nutno kontrolovat optimální chlazení svařování. Údržbáři musí být schopni rychle a přesně detekovat případné úniky chladicího systému, způsobené opotřebenými uzávěry elektrod či dalšími poruchami. Při výrobě automobilů v automatizovaných linkách roboti produkují až 40 vysoce přesných bodových svarů v extrémně krátkém časovém cyklu, a to ve 3 směnách po celý den. Pro rozptýlení tepla, vyprodukovaného vysokými proudy v procesu svařování je nutné chladit konce svorek elektrod, tzv. elektrodové čepičky. Měděné elektrody jsou navrženy jako součástky opotřebením a musejí být pravidelně vyměňovány. Bez chlazení konců elektrod, by však byly elektrody opotřebený mnohem rychleji. To by vedlo nejen k vyšším nákladům, ale také ke ztrátám výroby kvůli častým odstávkám pro údržbu zařízení. Aby se tomuto stavu předešlo, elektrodové čepičky při bodovém svařování jsou chlazeny šesti až osmi litry vody za minutu. Chlazení vody má teplotu mezi 20 až 40 °C a je dodáváno pod tlakem až 10 barů.[20]

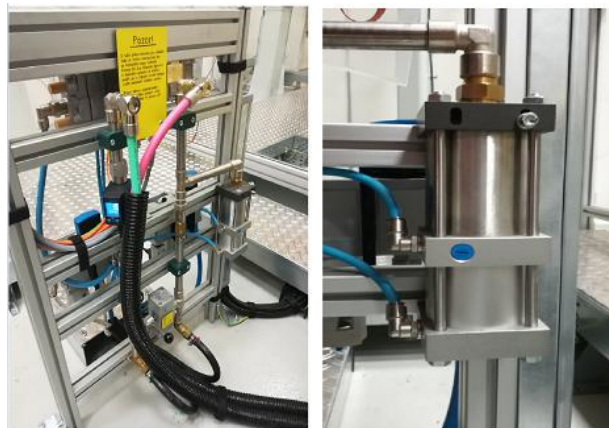
5.1 Poškození konektorů v dokovací spojkce vlivem vysokého tlaku

Jedním z problémů, který byl ve firmě Škoda Auto řešen, bylo poškození konektorů mezi dokovací spojkou a nástrojem robota v procesu aplikace kleští, tedy svařování. V případě tohoto procesu je nutno připojit oběh chladicí vody. Než dojde k tomuto kroku, je nutné odsát vodu mezi dokovací spojkou, (viz *Obrázek 19* – levá část) a RIP zařízením (viz *Obrázek 20* – levá část), které kontroluje a monitoruje stlačený vzduch a chladicí kapalinu dodávanou pro svařovací kleště při montáži karoserie. Modrá hadice je určena pro přívod vzduchu do kleští pod tlakem 6 barů. Zelená hadice slouží pro přívod chladicí vody do kleští a růžová hadice pro odvod ohřáté vody z kleští.



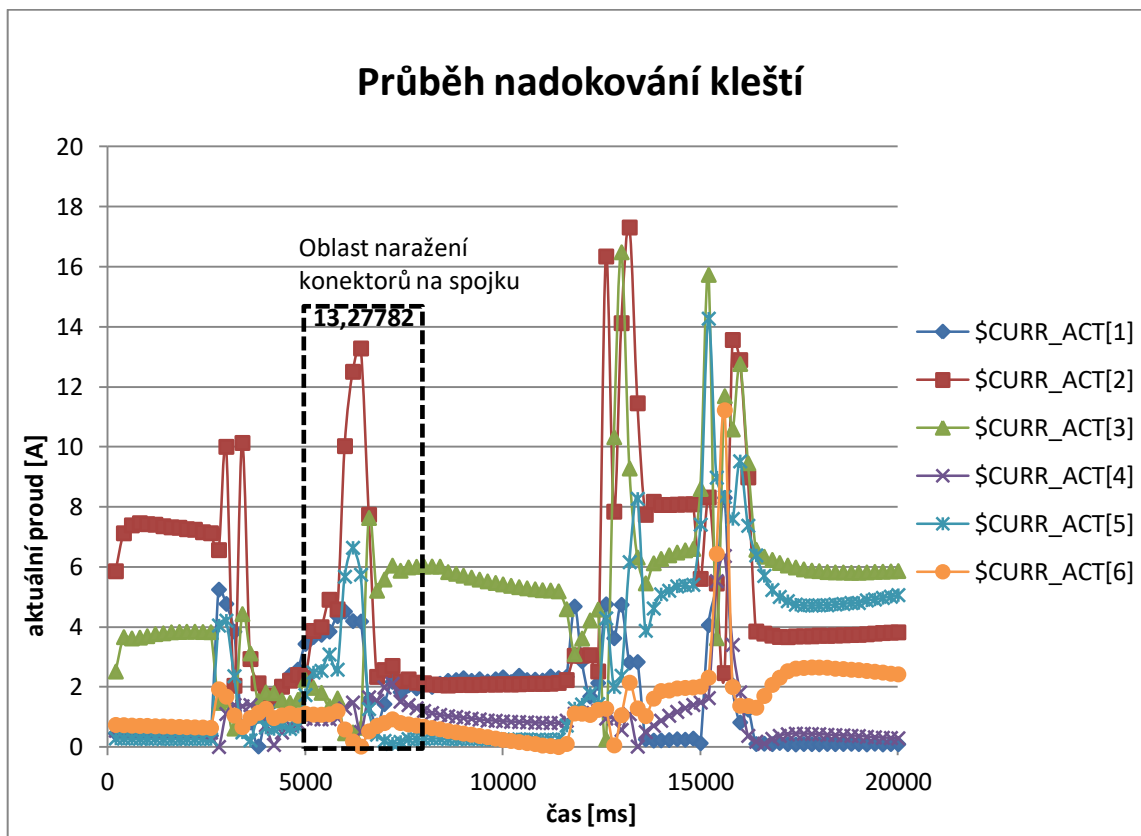
Obrázek 19: Dokovací spojka s křídélkem

Problém vznikal při nefunkčnosti odsávání vody do válce (viz *Obrázek 20* – pravá část). Po odpojení chladicího oběhu vody je nutno část zbylé vody mezi dokovací spojkou a RIP odstát do válce, tím se sníží tlak v této oblasti a dojde k optimálnímu nadokování kleští na robota. Pokud by nedošlo k odsátí vody, tak v dané oblasti zůstane vysoký tlak. V případě připojení kleští je poté nutno vyvinout větší sílu, než je optimální a tím dochází k poškození konektorů, resp. lámání křídélka (viz *Obrázek 19* – pravá část).



Obrázek 20: RIP zařízení a odsávací válec

Řešením tohoto problému je hlídání silových účinků během dokování kleští. Jedná se o měření proudů během dokování. Nejprve byl změřen ideální stav průběhu dokování a dle toho je určena odpovídající mez, kterou by robot poté neměl překročit (viz *Graf 1*). Průběh zobrazuje pohyb robota v automatickém režimu z výchozí polohy k dokovací spojce, pro napojení na kleště a zpět. Špičková hodnota proudu v okamžiku naražení byla v druhé ose, a to 13,27 A. Při opakovaném měření byla hodnota 13,21 A. Po nějaké době lze provést kontrolu a sledovat, jestli se hodnoty proudů v této fázi nezhoršují, resp. nezvyšují a nepůsobí na křídélko vysoké síly, které by vedly k jeho poškození.



Graf 1: Průběh nadokování kleští

5.2 Diagnostika signálů

Jedná se o testování komunikace signálů, tj. měření vstupů a výstupů, popř. flagů. Lze jím ověřit správnou komunikaci robota s technologií podle požadovaného schématu komunikace, tzv. impuls-plánu. Může se jednat např. o uvolňování nebo polohování kleští, zapnutí regulátoru, frézování a další technologie. Touto diagnostikou lze detekovat případný problém, které by mohl na pozadí narušovat či nevyžádaně ukončovat komunikaci, např. neočekávaný reset.

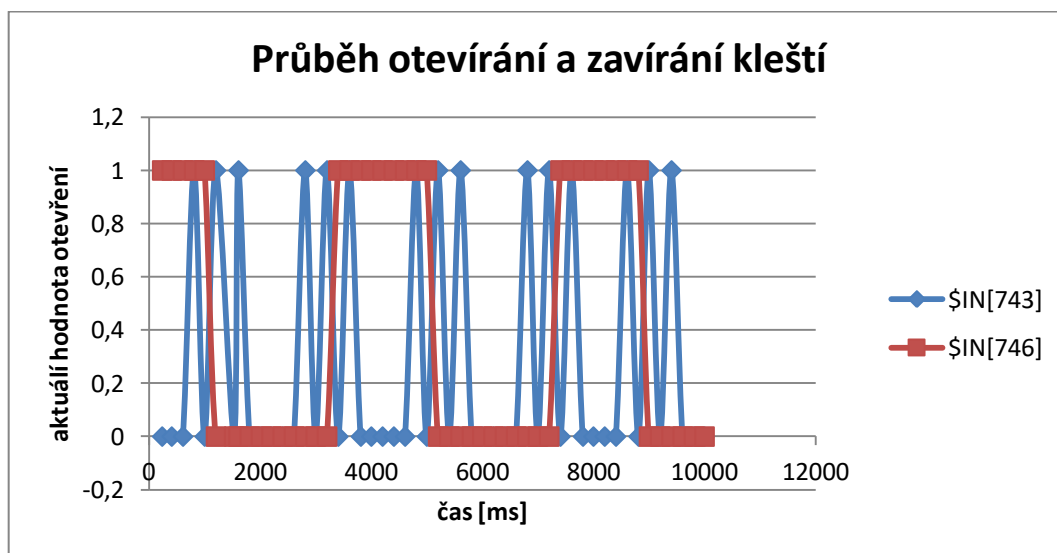
Teoretickým příkladem může být např. proměření signálů během otevírání/zavírání kleští a následné zjištění aktuální hodnoty. V příslušném standardu si uživatel nalezne

tabulku, kde si zjistí příslušnou adresu (rozsah) komunikace (viz *Obrázek 21*). Tabulka zobrazuje vstupy a výstupy, zakódování informace a komentář.

5.9 Kombi-Steuerung KE1 (Option B)							
E/A	Wort	Byte	Bit	Bezeichnung	Kommentar Eingaenge	Bezeichnung	Kommentar Ausgaenge
737	47	92	32	KOMBI 1	Istposition W1x0,1mm	KOMBI 1	Sollposition W1x0,1mm
738	47	92	33	EZ+SK	Istposition W2x0,1mm	EZ+SK	Sollposition W2x0,1mm
739	47	92	34		Istposition W4x0,1mm		Sollposition W4x0,1mm
740	47	92	35		Istposition W8x0,1mm		Sollposition W8x0,1mm
741	47	92	36		Istposition W16x0,1mm		Sollposition W16x0,1mm
742	47	92	37		Istposition W32x0,1mm		Sollposition W32x0,1mm
743	47	92	38		Istposition W64x0,1mm		Sollposition W64x0,1mm
744	47	92	39		Istposition W128x0,1mm		Sollposition W128x0,1mm
745	47	93	40	KOMBI 1	Istposition W256x0,1mm	KOMBI 1	Sollposition W256x0,1mm
746	47	93	41	EZ+SK	Istposition W512x0,1mm	EZ+SK	Sollposition W512x0,1mm
747	47	93	42		Istposition W1024x0,1mm		Sollposition W1024x0,1mm
748	47	93	43		Istposition W2048x0,1mm		Sollposition W2048x0,1mm
749	47	93	44		Istposition W4096x0,1mm		Sollposition W4096x0,1mm
750	47	93	45		Istposition W8192x0,1mm		Sollposition W8192x0,1mm
751	47	93	46		Istposition W16384x0,1mm		Sollposition W16384x0,1mm

Obrázek 21: Ukázka z impuls-plánu (rozměry otevření kleští)

Poté si v aplikaci zvolí požadované vstupy a výstupy z tabulky, v tomto případě rozsah 737 až 751. Každá komunikace s technologií má svoji specifickou adresu. Průběh zavírání a otevírání kleští zobrazuje graf v aplikaci (viz *Graf 2*). Vzhledem k vysokému počtu měřených hodnot (celkem 15), jsou v grafu pro přehlednost zobrazeny pouze 2 vstupy.



Graf 2: Průběh otevírání a zavírání kleští

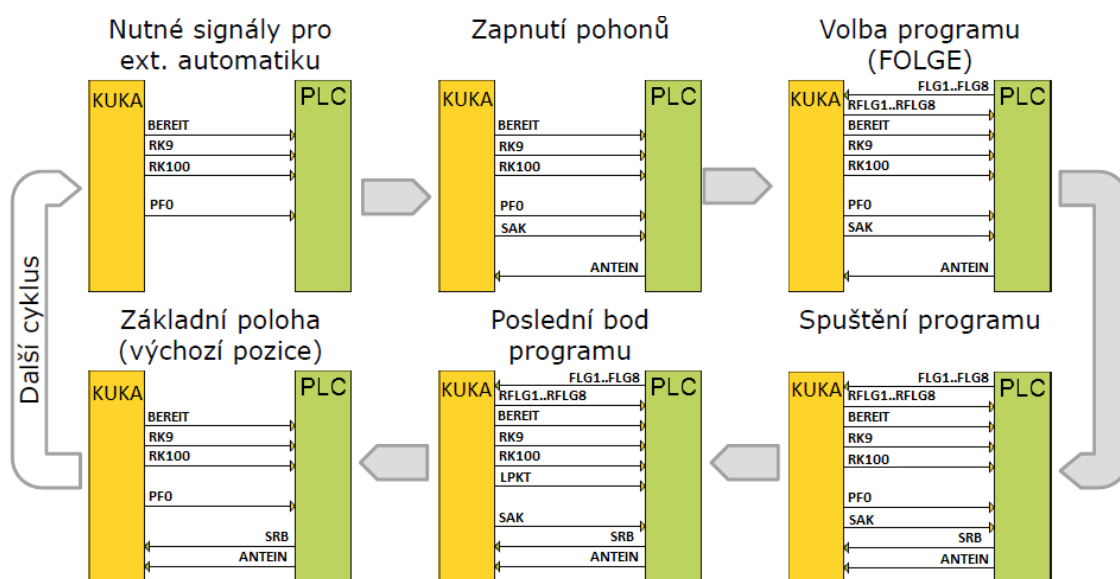
Z naměřených dat byly vypočteny změny aktuálních hodnot otevírání (viz *Tabulka 5*). Z naměřených dat a vypočtených hodnot lze ověřit požadovanou míru otevření kleští během sváření.

Tabulka 5: Aktuální hodnoty otevření kleští

Počet	1	2	3	4	5	6	7	8	9	10
Hodnota [mm]	68,3	68,1	66,2	60	57,6	48,1	25,9	36	30,2	27,8

Podobným příkladem může být např. proměření signálů v komunikaci mezi svařovacími kleštěmi ve svařovacím bodě a PLC. V příslušném standardu si uživatel opět nalezne tabulku, kde si zjistí adresu (rozsah) komunikace. Poté si v aplikaci zvolí požadované vstupy a výstupy z tabulky, v tomto případě rozsah 705 – 712. Následně může ověřit správný stav komunikace.

Dalším příkladem může být diagnostika externí automatiky podle určeného schématu (viz Obrázek 22). Pro vykonávání daného procesu robota je nutná výměna signálu mezi ním a PLC. Uživatel si v aplikaci navolí požadované vstupy a výstupy signálů, které jsou zakódované ve standardu. Následně lze měřením, resp. číslem programu (x), ověřit, zda komunikace externí automatiky mezi PLC a robotem proběhla správně.



Obrázek 22: Schéma externí automatiky

Robot je ve výchozí pozici (125). Poté je z PLC robotovi poslána na vstup binárně zakódovaná hodnota (124) s informací „robot do údržby“, tj. dojde k nastavení signálů pro externí automatiku – zapnutí pohonů – volba programu. Po vykonání těchto procesů se robot vrací do základní polohy (125). Hodnoty vstupů IN lze porovnat s výstupy OUT, jestli žádaný signál robot opravdu přijme. Na obrázku (viz Obrázek 23) lze vidět přechody mezi výchozí pozicí „home“ a údržbou „service“ z aplikace. Bylo změřeno (celkem 5x), že signály OUT odpovídají signálům IN se zpožděním (delay) cca 400 ms.

Toto zpoždění je především způsobeno na straně robota, a to zpracováním logického obvodu na pozadí. Dalším faktorem zpoždění je časová odezva profinetu, která je pro standardní komunikaci (datové přenosy TCP/IP) 100 či více ms. Položka IN[9] značí signalizaci spuštění programu, tzv. „folgestart“, ve schématu znázorněno jako SRB.

t [ms]	\$OUT[1]	\$OUT[2]	\$OUT[3]	\$OUT[4]	\$OUT[5]	\$OUT[6]	\$OUT[7]	\$OUT[8]	\$IN[1]	\$IN[2]	\$IN[3]	\$IN[4]	\$IN[5]	\$IN[6]	\$IN[7]	\$IN[8]	\$IN[9]
203	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
406	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
609	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
813	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
1000	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
1203	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
...	FALSE
8813	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9000	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9203	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9406	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9609	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
9813	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
10000	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
	HOME	DELAY	SERVICE	folgestart													

Obrázek 23: Odezva výstupů Folge

5.3 Praskání greiferu vlivem přetížení

Tento příklad se týká zatížení nástroje. V závodě v Kvasinách docházelo k praskání greiferů vlivem přetížení (viz Obrázek 24). Za normálních okolností nelze zjistit průběh polohy, popř. rychlosti pohybu nástroje.



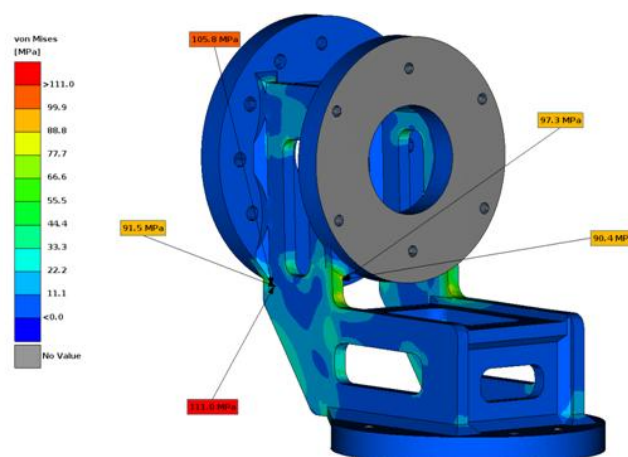
Obrázek 24: Prasknutý greifer [21]

Pomocí softwaru lze změřit reálný pohyb robota s nástrojem v kartézských souřadnicích. Následně jsou předána naměřená data výpočtářům, kteří pomocí pevnostní analýzy, tj. simulace a výpočtů ve speciálním softwaru dokážou získat maximální zatížení greiferu pro konkrétní případ. Oprava, seřízení greiferu a korekce všech drah robota způsobila zpoždění a ztráty uvedené v záznamu o poruše [21]. Pomocí žádaných dat z robota, tj. pohybových map robota, nutných pro analýzu zatížení greiferu, se lze vyvarovat následujícím ztrátám, které byly způsobeny (viz Tabulka 6).

Tabulka 6: Dopady prasklého greiferu ve výrobě

Následky prasklého greiferu	
Prostoj:	435 minut
Diagnostika:	15 minut
Ztráta:	50 karoserií

Na obrázku (viz *Obrázek 25*) je zobrazen vytvořený model greiferu Heckklape SK482 s hodnotami hraničních napětí, které na něj působí. Obrázek je převzat z výsledné prezentace od společnosti, která tento problém pro Škoda Auto řešila. Nazývá se IDIADA CZ a sídlí v Liberci. Zabývá se oblastí designu, inženýringu, testování a homologace v automobilovém průmyslu. Závěr z výsledné prezentace zněl [22]: “*Výpočet proveden pouze se statickým zatížením. Domníváme se, že v důsledku pohybu robota působí na konstrukci další síly přispívající k poškození. Nutné zahrnout do výpočtu. Prosíme o dodání pohybových map robota.*“



Obrázek 25: Výsledky napětí metodou Von Mises [22]

5.4 Další způsoby využití

Dalším teoretickým příkladem, který byl v podniku řešen, byla aplikace broušení. V tomto procesu je důležité dodržet optimální obvodovou rychlost. Během broušení dochází k obrušování kotouče a zmenšování jeho poloměru, je nutno tedy zvyšovat otáčky pro dodržení optimální rychlosti. Pomocí vyvinuté aplikace lze diagnostikovat, zdali ke zvyšování otáček opravdu dochází a jak se hodnota v průběhu navyšování otáček případně mění. V neposlední řadě lze pomocí softwaru detekovat teplotu motorů uvnitř robota, zdali nedochází k přehřívání.

6 Návrhy na vylepšení SW

Aplikace by se mohla vylepšit např. převedením výpočtu přímé úlohy do procesu měření polohy kartézských souřadnic. Tím by došlo k ušetření vzorkovacího času i počtu měřených proměnných. Stačilo by měřit pouze 6 kloubových proměnných a následně pomocí naprogramované metody přepočítávat. Tím se získají kartézské souřadnice (x, y, z) a úhly otočení kolem jejich os (a, b, c) .

Dále by se mohl realizovat seznam nejpoužívanějších diagnostických měření, která by uživatel pouze vybral ze seznamu, a tím se vyhnul vybírání jednotlivých prvků do seznamu instrukcí.

7 Závěr

Hlavním cílem této diplomové bylo vytvořit aplikační software pro robota KUKA KR210 R2700 EXTRA C4, který bude schopen sbírat technologická data z robota (měřené veličiny) a ukládat je do souboru v dohodnuté formě. Data by měla být následně vizualizovaná vhodným způsobem pro jejich prezentaci.

Tento software byl vyvinut pro společnost Škoda Auto, resp. pro oblast pro svařovny, kde se bude používat jako diagnostická aplikace při řešení poruchových stavů či ověření správné komunikace mezi technologií a robotem.

Komunikace mezi robotem a PC byla uskutečněna na bázi komunikačního protokolu TCP a byla zajištěna prostřednictvím průmyslové sběrnice profinet. Součástí komunikace bylo použití software KukavarProxy, který pomohl zprostředkovat získávání dat na straně robota.

V rámci zadání se povedlo vytvořit aplikaci, která umožňuje sběr dat z jakéhokoliv robota značky KUKA ve svařovně. Aplikace obsahuje komunikační rozhraní pro připojení k robotu a lze pomocí ní měřit, ukládat a vizualizovat veličiny uvedené v seznamu. Z hlediska využití práce jsou v seznamu uvedeny jen důležité veličiny. Teoreticky lze měřit jakoukoliv veličinu z robota, která je zakódovaná v seznamu instrukcí. Součástí měření je i volba grafu, který přehledně zobrazuje křivku naměřených hodnot. Aplikace obsahuje okno na výpočet kartézských souřadnic robota a úhlů natočení kolem jejich os, z předem vyplněné Denavit-Hartenbergovi tabulky. Aplikace bude primárně používána pro sběr dat a jako diagnostika před možným poškozením jednotlivých zařízení. Jako příklad je uvedena diagnostika poškození konektorů v dokovací spojce, což je v práci popsáno a vysvětleno na příslušném grafu nebo jako součást řešení analýzy zatížení greiferu.

Podařilo se splnit všechny body zadání diplomové práce, vytvořit aplikaci pro sběr a reprezentaci dat z robota, která bude sloužit pro optimalizaci robotických aplikací, jak je uvedeno na několika konkrétních příkladech.

Seznam použité literatury

- [1]. *Od 1. průmyslové revoluce ke 4.: Technický portál.cz* [online]. Praha: Business Media CZ, 2015 [cit. 2018-04-06]. Dostupné z: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html
- [2]. *Teslův transformátor*. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 4.10.2017 [cit. 2018-03-09]. Dostupné z: https://cs.wikipedia.org/wiki/Tesl%C5%AFv_transform%C3%A1tor
- [3]. *Přístroje pro měření a metrologii. PRIMA BILAVČÍK, s.r.o.* [online]. Uherský Brod, 2014 [cit. 2018-03-09]. Dostupné z: <http://www.merici-pristroje.cz/novinky/detail/prumysl-4-0-otazky-a-odpovedi.htm>
- [4]. *Iniciativa Průmysl 4.0. Ministerstvo průmyslu a obchodu* [online]. Praha, 2017 [cit. 2018-03-09]. Dostupné z: <https://www.mpo.cz/cz/rozcestnik/ministerstvo/aplikace-zakona-c-106-1999-sb/informace-zverejnovane-podle-paragrafu-5-odstavec-3-zakona/-iniciativa-prumysl-4-0--230485/>
- [5]. *Robotizaci se nevyhne ani český průmysl. Investiční web: Robotizaci se nevyhne ani český průmysl* [online]. Praha: Investičníweb, 2017, 28.9.2017 [cit. 2018-03-22]. Dostupné z: <http://www.investicniweb.cz/news-robotizaci-prumyslu-se-nevyhne-ani-cesky-prumysl/>
- [6]. *Hospodářské noviny* [online]. Praha: Economia, 2018 [cit. 2018-04-06]. Dostupné z: <https://byznys.ihned.cz/c1-66084510-samoriditelne-auto-poprvezabilo-chodce-spolecnost-uber-kttere-vuz-patril-kvuli-nehode-pozastavuje-vsechny-testy-autonomni-jizdy>
- [7]. *KR QUANTEC extra* [online]. Praha: KUKA Roboter CEE [cit. 2017-09-30]. Dostupné z: <https://www.kuka.com/cs-cz/produkty,-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/pr%C5%AFmyslov%C3%A9-roboty/kr-quantec-extra>

- [8]. KUKA: Download Center. *KUKA: Download Center* [online]. KUKA Aktiengesellschaft [cit. 2017-09-30]. Dostupné z: https://www.kuka.com/cs-cz/services/downloads?terms=product_name:KR%20210%20R2700%20extra
- [9]. *Produktivita.cz: Standardizace* [online]. Ing. Otakar Ježek, 2006 [cit. 2018-04-06]. Dostupné z: <http://www.produktivita.cz/cs/metody-prumysloveho-inzenyrstvi/standardizace.html>
- [10]. *MRBT – Robotika: M6. Doplnění encyklopedie Wikipedia - Kinematika robotů* [online]. Brno: Martin Kozel, 2013 [cit. 2017-09-27]. Dostupné z: http://www.uamt.feec.vutbr.cz/~robotika/2013_MRBT/2013_M06_kinematika.pdf
- [11]. Figure 1: Denavit-Hartenberg kinematic parameters. In: *Researchgate* [online]. 2013 [cit. 2017-10-02]. Dostupné z: https://www.researchgate.net/figure/257067449_fig1_Figure-1-Denavit-Hartenberg-kinematic-parameters-4
- [12]. RICHARD, Paul. *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators* [online]. England: Cambridge, MA: MIT Press, 1981 [cit. 2018-04-06]. ISBN 978-0-262-16082-7. Dostupné z: <https://books.google.cz/books?id=UzZ3LAYqvRkC&printsec=frontcover&redirect=y#v=onepage&q&f=false>
- [13]. *Roboty a manipulátory: Učební text VOŠ a SPŠ Kutná Hora* [online]. Kutná Hora: Učební text VOŠ [cit. 2017-09-27]. Dostupné z : http://www.edumat.cz/texty/Roboty_manipulatory.pdf
- [14]. Forward Kinematic Analysis of an Industrial Robot: New Developments in Mechanics and Mechanical Engineering. In: CONSTANTIN, Daniel, Dan-Ilie BULIGA, Marin LUPOAE a Cătălin BACIU. *Forward Kinematic Analysis of an Industrial Robot* [online]. s. 1-6 [cit. 2017-09-27]. ISBN 978-1-61804-288-0. Dostupné z: <http://www.inase.org/library/2015/vienna/bypaper/MECH/MECH-13.pdf>

- [15]. ČERNOHORSKÝ, Josef. *ZÁKLADY ROBOTIKY: Denavit-Hartenbergova transformace* [online]., 1-37 [cit. 2017-09-27]. Dostupné z: <http://www.fm.tul.cz/esf0247/index.php?download=376>
- [16]. Robotika: Teorie průmyslových robotů. *SlidePlayer* [online]. Brno, 2016 [cit. 2017-10-07]. Dostupné z: <http://slideplayer.cz/slide/1885278/>
- [17]. SLABAUGH, Gregory G. Computing Euler angles from a rotation matrix. In: *Citeseerx.ist.psu.edu* [online]. [cit. 2018-01-18]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.371.6578&rep=rep1&type=pdf>
- [18]. KUKA System Software (KSS): Systémové proměnné. 03. 2005.
- [19]. KUKAVARPROXY 在 KRC4 中接收和发送消息的格式 (1). *LT: Revolution road* [online]. 2014 [cit. 2018-03-31]. Dostupné z: http://davidrobot.com/2014/09/kukavarproxy_krc4_msg_format_1.html
- [20]. Monitored cooling for robot spot welding units. *Christian Bürkert GmbH & Co. KG* [online]. Ingelfingen, 2011 [cit. 2018-03-27]. Dostupné z: <https://www.burkert.com/en/Company-Career/What-s-New/Press/Media/Technical-Reports/Technical-Reports-additional-topics/Monitored-cooling-for-robot-spot-welding-units>
- [21]. Interní dokumenty ŠKODA AUTO a.s.: Záznam o poruše zařízení. Mladá Boleslav, 2017.
- [22]. ALBRECHT, David. Pevnostní analýza greiferu Heckklappe SK482. Liberec, 2017.
- [23]. Interní dokumenty ŠKODA AUTO a.s.: Svařovna SK370/X - workshop. Mladá Boleslav, 2017

A Příloha – Obsah přiloženého CD

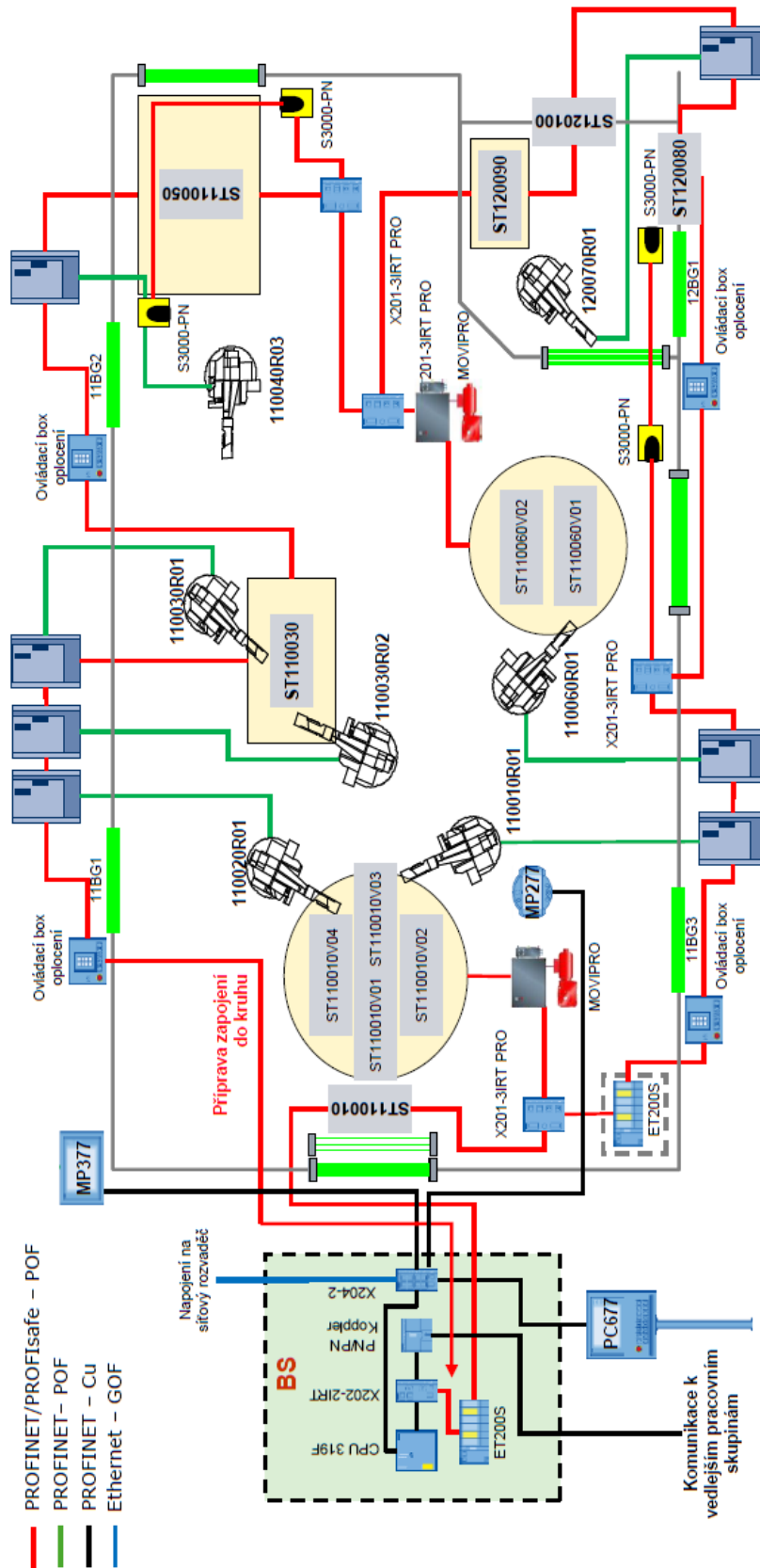
- text diplomové práce
 - diplomová_práce_2018_Jindrich_Titlbach.pdf
- zdrojový kód programu
 - aplikace pro PC (v programovacím jazyce C#)

B Příloha – Zdrojový kód pro převod na aktuální proud

```
private double GetCurrent(int Index)//prevod na akt proud
{
    string s = readVar("$CURR_ACT[" + Index + "]");
    Regex regex = new Regex(@"[-+]?([0-9]*\.[0-9]+|[0-9]+)");
    Match match = regex.Match(s);
    string s1 = match.Value;
    if (s.Contains("E-0")) s1 += s.Substring(s.IndexOf("E-0"), 4);
    double RealValue = double.Parse(s1, CultureInfo.InvariantCulture);
    RealValue = RealValue / 100;
    double MaxCurrent = 40; //hodnota ziskana z dokumentace menice
    RealValue = Math.Abs(RealValue) * MaxCurrent;
    return RealValue;
}
```


C.1 Příloha – Standard topologie pro zasíťování

Obrázek [23] zobrazuje topologii pro zasíťování svařovny SK370/x.



C.2 Příloha – Standard pro označování profinetu

Obrázek [23] zobrazuje příklad pro formát označování profinetu (sběrnice).

