



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## ŘÍZENÍ PRŮMYSLOVÉHO ROBOTA

INDUSTRIAL ROBOT CONTROLLING

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ZUZANA STUDENÁ

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. RADEK KOČÍ, Ph.D.

BRNO 2016

## **Zadání bakalářské práce**

Řešitel: **Studená Zuzana**  
Obor: Informační technologie  
Téma: **Řízení průmyslového robota**  
**Industrial Robot Controlling**

Kategorie: Softwarové inženýrství

### **Pokyny:**

1. Seznamte se s problematikou kolaborativních robotů. Zpracujte přehled aktuálního stavu v této oblasti.
2. Navrhněte řízení kolaborativního robota, který bude na základě podnětů od člověka měřit kvalitu silových a momentových změn v mechanických pohyblivých částech automobilu.
3. Vytvořte aplikaci klient-server pro řízení kolaborativního robota.
4. Aplikaci rozšiřte o možnost vizualizace naměřených sil a momentů. Data pro vizualizaci budou získávána z databáze historických dat.
5. Aplikaci otestujte na reálném zařízení.
6. Diskutujte přínos výsledků práce a možnosti dalšího vývoje v oblasti kolaborativní robotiky v průmyslu.

### **Literatura:**

- Podle pokynů vedoucího práce.

Pro udělení zápočtu za první semestr je požadováno:

- První 3 body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Kočí Radek, Ing., Ph.D.**, UITS FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav inteligentních systémů  
602 00 Brno, Božetechova 2

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## Abstrakt

Cielom práce je vytvoriť softvér pre zber a vizualizáciu dát nameraných pri automatizovanom testovaní mechanických častí automobilov. Mechanické časti sú testované kolaboratívnym robotom LBR iiwa, ktorý posiela namerané údaje desktopovej aplikácii. Práca popisuje návrh a implementáciu aplikácie pre riadenie robota a aplikácie pre zber a vizualizáciu nameraných dát. V poslednej časti sa práca zaoberá testovaním výsledného produktu na reálnom zariadení a diskutuje možnosť nasadenia v praxi.

## Abstract

The aim of this work is to create software for the collection and visualization of data measured during automated testing of mechanical parts for cars. Mechanical parts are tested by collaborative robot LBR iiwa, which sends the measured data to desktop application. The work describes the design and implementation of application for controlling the robot, collection and visualization of measured data. Last part of the work deals with testing of the final product on a real device and discusses the possibility of deployment in practice.

## Klíčové slová

Kolaborativita, Robotika, Kolaboratívny robot, Riadenie robota, LBR iiwa, Zber dát, Vizualizácia dát, Testovanie automobilových častí, Java, Industry 4.0

## Keywords

Colaboration, Robotics, Collaborative robot, Robot control, LBR iiwa, Data collection, Data visualization, Testing automotive components, Java, Industry 4.0

## Citácia

STUDENÁ, Zuzana. *Řízení průmyslového robota*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Kočí Radek.

# Řízení průmyslového robota

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracovala samostatne pod vedením pána Ing. Radka Kočího, Ph.D. Uviedla som všetky literárne parametre a publikácie, z ktorých som čerpala.

.....  
Zuzana Studená  
17. mája 2016

## Podakovanie

Chcela by som sa poďakovať vedúcemu mojej práce Ing. Radkovi Kočímu, Ph.D. za poskytnutý čas a odbornú pomoc, ktoré mi pomohli viesť moju prácu správnym smerom. Ďalej by som sa chcela poďakovať Ing. Marošovi Mudrákovi a Ing. Stanislavovi Korcovi za ich čas a pomoc pri vývoji práce. Na záver ďakujem spoločnosti Matador Industries za poskytnutie robota pre vypracovanie práce.

© Zuzana Studená, 2016.

*Táto práca vznikla ako školské dielo na FIT VUT v Brně. Práca je chránená autorským zákonom a jej využitie bez poskytnutia oprávnenia autorom je nezákonné, s výnimkou zákonne definovaných prípadov.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Aktuálny stav robotiky</b>	<b>4</b>
2.1 Dôležité pojmy . . . . .	4
2.2 Rozdelenie robotiky . . . . .	5
2.2.1 Humanoidná robotika . . . . .	5
2.2.2 Priemyselná robotika . . . . .	5
2.3 Vývoj priemyselnej robotiky . . . . .	6
2.3.1 Počiatok priemyselnej robotiky . . . . .	6
2.3.2 Hlavný rozvoj . . . . .	6
2.3.3 Súčasnosc a budúcnosť . . . . .	7
2.4 Rozdelenie robotov . . . . .	8
2.4.1 Konvenčné roboty . . . . .	8
2.4.2 Kolaboratívne roboty . . . . .	8
2.4.3 Prehľad kolaboratívnych robotov . . . . .	9
<b>3 Špecifikácia robota a vývojového prostredia</b>	<b>15</b>
3.1 Robot LBR iiwa . . . . .	15
3.2 Sunrise Wokbench . . . . .	15
3.3 Konfiguračné nastavenia robota . . . . .	16
3.3.1 Súradný systém . . . . .	16
3.3.2 Preddefinované frame-i . . . . .	16
3.3.3 Preddefinované nástroje . . . . .	16
3.3.4 Monitorovacie mechanizmy . . . . .	16
3.3.5 Pohyby . . . . .	16
3.3.6 Zaznamenávanie dát . . . . .	17
3.3.7 Pohybové podmienky . . . . .	17
3.4 Komunikácia s okolím . . . . .	17
<b>4 Analýza a návrh práce</b>	<b>18</b>
4.1 Analýza robotickej aplikácie . . . . .	18
4.2 Návrh robotickej aplikácie . . . . .	18
4.3 Analýza desktopovej aplikácie . . . . .	19
4.4 Návrh desktopovej aplikácie . . . . .	21
4.5 Návrh komunikácie . . . . .	22

<b>5 Implementácia</b>	<b>23</b>
5.1 Robotická aplikácia . . . . .	23
5.1.1 Beh aplikácie . . . . .	23
5.2 Desktopová aplikácia . . . . .	24
5.2.1 Hlavná časť . . . . .	26
5.2.2 Grafické rozhranie . . . . .	26
5.2.3 Riadenie aplikácie . . . . .	27
5.2.4 Komunikácia . . . . .	28
<b>6 Testovanie</b>	<b>29</b>
6.1 Testy nepriaznivých vplyvov na meranie . . . . .	29
6.1.1 Správanie pri kontakte . . . . .	30
6.1.2 Prázdne stlačenie robota . . . . .	31
6.1.3 Prekážka pri stlačení . . . . .	31
6.2 Testy tlačidiel palubnej dosky . . . . .	32
6.2.1 Testovanie tlačidla ručnej brzdy . . . . .	32
6.2.2 Testovanie tlačidiel auto-hold a lock/unlock . . . . .	33
6.3 Záver testovania . . . . .	33
<b>7 Záver</b>	<b>35</b>
<b>Literatúra</b>	<b>36</b>
<b>Prílohy</b>	<b>38</b>
Zoznam príloh . . . . .	39
<b>A Obsah DVD</b>	<b>40</b>
<b>B Záznam o chybnom meraní vytvorený v desktopovej aplikácii</b>	<b>41</b>
<b>C Užívateľský Manuál</b>	<b>42</b>
C.1 Požiadavky aplikácie: . . . . .	42
C.2 Preklad a spustenie . . . . .	42
C.3 Ovládanie . . . . .	42

# Kapitola 1

## Úvod

V dnešnej dobe roboty uľahčujú ľudskú prácu takmer v každom odvetví. Asi najpodstatnejšiu úlohu plnia roboty práve v priemysle. V automatizačnom priemysle dnes pracuje približne trisťtisíc robotov po celom svete. S vývojom inteligentných tovární prichádza na trh nový druh robotov nazvaných kolaboratívne roboty. Jedná sa o nový druh spolupráce medzi človekom a robotom, ktorý nieje v súčasnej dobe príliš rozšírený a hromadne nasadzovaný. Je nutné vymýšľať nové aplikácie a spôsoby uplatnenia týchto robotov. Z tohoto dôvodu som sa na podnet spoločnosti Matador Industries rozhodla túto prácu venovať prieskumu v oblasti kolaboratívnych robotov a hlavne vývoju aplikácie pre kolaboratívneho robota.

Práca je rozdelená na teoretickú a praktickú časť. Teoretická časť práce je zameraná na prieskum aktuálneho stavu robotiky, postupného vývoja a vzniku kolaboratívnych robotov. V praktickej časti sa zaoberám návrhom a implementáciou riadiacej aplikácie pre robota LBR iiwa a desktopovej aplikácie bežiacej na platforme Windows. Robotická aplikácia riadi robota LBR iiwa, ktorého úlohou je na podnet pracovníka testovať mechanické časti v automobiloch. Pri testovaní sú zaznamenávané sily pôsobiace na testované mechanické časti, ktoré sú následne posielané desktopovej aplikácii. Desktopová aplikácia prijíma, ukladá a zobrazuje prijaté hodnoty. Užívateľ má možnosť zobrazíť prípadne exportovať aktuálne alebo historicky namerané hodnoty testovaných častí. Na základe týchto hodnôt dokáže užívateľ rozhodnúť o kvalite testovaného zariadenia. Aplikácia ako celok by mala uľahčiť, prípadne úplne nahradiť činnosť človeka pri testovaní mechanických komponentov.

Teoretická časť práce je popísaná v kapitole 2. Táto kapitola slúži ako úvod od problematiky kolaboratívnych robotov. Zachytáva vývin priemyselnej robotiky od jej počiatku až po súčasnosť, popisuje vlastnosti a rozdiely medzi klasickými konvenčnými a novými kolaboratívnymi robotmi. V poslednej sekcii sú popísané najznámejšie kolaboratívne roboty. Kapitola 3 sa špecializuje na podrobný popis robota, pre ktorého je vytváraná cieľová aplikácia. Kapitoly 4 a 5 popisujú priamo analýzu, návrh a riešenie daného problému.

## Kapitola 2

# Aktuálny stav robotiky

### 2.1 Dôležité pojmy

Ako bolo spomenuté v úvode, časť práce je zameraná na preskúmanie aktuálneho stavu robotiky, ktorá umožňuje spoluprácu človeka a robota na zdieľanom pracovisku. Z tohoto dôvodu, považujem za dôležité uviesť definíciu a význam nasledovných pojmov v priemyselnej robotike.

**Koexistovať.** *Byť (existovať) závisle alebo nezávisle na jednom mieste v rovnakom čase [2].*

V súvislosti s priemyselnou robotikou pojem **koexistovať** resp. **koexistujúci** chápeme ako možnosť existencie pracovníka a robota na zdieľanom pracovisku bez porušenie stanovených bezpečnostných noriem.

**Kooperatívny.** *Slovo možno odvodiť od slova **kooperácia** - spolupráca viacerých strán za zámerom spoločného úspechu, spoločná práca prospešná obom stranám [2].*

V robotike pojem **kooperácia** možno vysvetliť rôznymi spôsobmi:

1. forma interakcie založená na komunikácii [23],
2. kolaboratívne pôsobenie zamerané na dosiahnutie spoločného cieľu,
3. spolupráca za účelom zvýšenia výkonnosti alebo úspory času [17].

**Kolaboratívny.** *Ovodené od slova **kolaborativita** - inteligentná spolupráca človeka a robota [14]. Človek a robot zdieľajú spoločné pracovisko.*

V robotike je slovo **kooperujúci** úzko spojené so slovom **kolaboratívny**. Ide o pojmy popisujúce spoločnú spoluprácu na robotickom pracovisku. Môže sa jednať o vzdialenú spoluprácu, prostredníctvom vzdialeného navádzania robota. Alebo priamu spoluprácu, kedy sa človek a robot dostávajú do priameho kontaktu. Zatiaľ čo robot vykonáva stereotypné činnosti, človek sa môže venovať činnostiam, ktoré nedokáže robot vykonať. V prípade priameho kontaktu musí byť robot schopný svoju činnosť spomaliť alebo zastaviť bez ohrozenia bezpečnosti a zdravia pri práci.



## 2.2 Rozdelenie robotiky

V dnešnej dobe je robotika nenahradiiteľnou súčasťou ľudského života. S jej vývojom neustále narastá rozsah uplatnenia. Roboty nahrádzajú ľudskú činnosť v priemysle, zdravotníctve, doprave, vede a mnohých ďalších odvetviach. Dôvody hromadného nasadzovania robotov sú prevažne ekonomického charakteru. Neopomenuteľnou výhodou je efektivita práce. S využitím robotiky možno nahradiť stereotypné činnosti človeka, zvýšiť kvalitu a presnosť práce [16].

Podľa možnosti uplatnenia (nasadenia) sa robotika delí do rôznych skupín. V rôznych literatúrach sa názory na rozdelenie robotiky líšia v závislosti od autora textu. Medzi najviac rozvinuté odvetvia možno zaradiť priemyselnú robotiku, servisnú alebo humanoidnú robotiku, vesmírnu robotiku, medicínsku, vzdelávaciu a iné.

### 2.2.1 Humanoidná robotika

**Humanoid:** Robot napodobňujúci človeka fyzickým vzhľadom a zároveň mentálnym správaním. Telo robota pozostáva z trupu, hlavy a končatín.

Cielom humanoidnej robotiky je vytváranie robotov, ktorých vzhľad, správanie a myslenie je blízko inšpirované človekom. Robot by mal napodobiť pohyby človeka ale aj vnímanie sveta a reakcie na podnety z okolitého prostredia. Dôvod vývoja humanoidnej robotiky vznikali z dvoch hlavných dôvodov:

- vytvoriť robota, ktorý bude nahrádzať ľudskú činnosť (otroka, asistenta)
- vytvoriť robota spoločníka

V druhej polovici dvadsiateho storočia pokrok v počítačovom priemysle umožnil ďalší rozvoj humanoidnej robotiky. Prvý robot, ktorý bol schopný samostatného pohybu, manipulácie s predmetmi a zároveň vnímania okolia bol WOBOT-1. WOBOT-1 bol vyvinutý Japonskou Univerzitou Waseda v roku 1973 [20].

Vývoj prebiehal aj na ďalších svetových univerzitách. Univerzita MIT v roku 1993 vytvorila robota Cog, ktorý sa učil z vykonávaných pohybov. V nasledujúcich rokoch sa objavovali ďalší roboti od iných veľkých spoločností. Známi sa stal robot Asimo od spoločnosti Honda. V posledných rokoch je kladený veľký dôraz na rozvoj humanoidnej robotiky. Na trhu sa objavujú stále noví roboti, ktorí dokonalejšie napodobujú ľudské správanie. Niektoré sú zamerané na pohybu v teréne, niektoré na spoluprácu s človekom. Rozpoznávanie hlasu, emócií, učenie na základne získaných poznatkov tvoria ďalšiu časť vývoja humanoidnej robotiky. Uplatnenie nachádzajú v zdravotníctve, vzdelávaní, zábave, ako spoločníci, pomocníci v domácnostiach a mnohé ďalšie [12].

### 2.2.2 Priemyselná robotika

Narozdiel od humanoidných robotov, priemyselné roboty sa nesnažia napodobniť človeka. Väčšinou sa jedná o programovateľné manipulátory. Pomocou presne naprogramovanej sekvencie pohybov manipulátory dokážu pohybovať predmetmi alebo vykonávať iné činnosti napr. zváranie, rezanie, zalievanie, povrchové úpravy materiálov [16].

Jedná sa väčšinou o jedno alebo dvoj-ramenové zariadenia pohybujúce sa v troch alebo viacerých osiach. Inou vlastnosťou priemyselných robotov je stupeň voľnosti. Stupeň voľnosti zaručuje robotu voľnosť a rozsah jeho pohybu. Aby bol robot schopný pokryť ľubovoľný bod v priestore potrebuje nadobúdať tretí stupeň voľnosti.

Väčšina priemyselných robotov je umiestnená za bezpečnostnými zábranami alebo v kliebkach. Súčasný vývoj bezpečnostných prvkov umožňuje bezbarierovú spoluprácu človeka s robotom. Vzniká nový pojem kolaboratívna spolupráca (spomenuté v 2.1) človeka a robota [17].

Hlavné uplatnenie priemyselnej robotiky je v automaticázii a výrobnom priemysle. Priemyselné roboty sú navrhnuté na stereotypné činnosti, ktoré nepotrebujú veľký rozsah pôsobenia. Väčšinou ide o automatizované linky, ktoré cyklicky vykonávajú jednu činnosť. Roboty zabezpečujú väčšiu rýchlosť, presnosť ale aj výslednú kvalitu práce.

## 2.3 Vývoj priemyselnej robotiky

Robotika je súčasťou ľudského života už viac ako šesťdesiat rokov. Rôzne zdroje [25, 19] považujú jej vývoj za tretiu etapu priemyselnej revolúcie. Najväčší rozvoj nastal v 70. rokoch 20. storočia a pretrváva dodnes. V tejto sekcii je popísaný rozvoj priemyselnej robotiky od jej počiatku až po súčasnosť.

### 2.3.1 Počiatok priemyselnej robotiky

Za otca robotiky sa pokladá George Devol ,ktorý v roku 1956, spolu Jozefom Engelbergerom založili prvú robotickú firmu Universal Animation. Podľa patentu Devola vytvorili v roku 1959 prvé robotické rameno UNIMATE. Robot vážil viac ako dve tony a bol riadený podľa programu na magnetickej páske.

Prvé roboty UNIMATE boli inštalované v roku 1961 vo firme General Motors. Roboty boli používané na striekanie karosérii automobilov. V roku 1969 firma General motors ako prvá tieto roboty využila na zváranie karosérií. Použitie robotov umožnilo automatizáciu zvárania približne na 90-tich percentách karosérie.

### 2.3.2 Hlavný rozvoj

Podľa článku [25] najväčší rozvoj nastal v 70. rokoch. Firma Kuka v roku 1973 vyvinula prvého robota, ktorý mal šesť elektronicky riadených osí. Richard Hohn v roku 1974 vyvinul prvého robota, riadeného pomocou mikropočítača. Robot mal názov T3 (The Tomorrow Tool). V tom istom roku firma Kawasaki použila Unimate roboty na zváranie kostry motocyklov. Vznikol prvý mikroprocesorom riadený robot.

Od roku 1975 začali na trh prichádzať montážne roboty. Robot PUMA od firmy Unimation/Vican bol prispôsobený na montáž malých dielov. 1978 Hiroshi Makino z Japonska vytvoril robota SCARA (Selective Compliance Assembly Robot Arm) . SCARA bolo dvoj-kĺbové robotické rameno podobné ľudskej paži. Vďaka jedinečnej konštrukcii motorov v jednotlivých kĺboch SCARA dosiahol malú tuhosť v osiach X,Y a veľkú v osy Z. Takto navrhnutý dizajn zaručoval lepšiu pohyblivosť v priestore, umožnil prístup do ťažko dostupných miest a zvýšil rýchlosť a precíznosť práce. V Japonsku nahradili hydraulické riadenie elektromotorom. Využitie elektrických motorov umožnilo odstrániť hydrauliku (čerpádlá, rozdeľovače), a tým zjednodušil celý mechanizmus a zároveň zachovať presnosť.

Prevrat v roku 1985 dosiahla spoločnosť KUKA, ktorá navrhla rameno v tvare Z. Dizajn zmenšil požiadavky na veľkosť potrebného priestoru na robotickom pracovisku. Pomocou troch posuvných a troch rotačných pohybov rameno dosiahlo celkom až šesť stupňov voľnosti. 1994 firma Motoman predviedla prvý MRC (Multy Robot Control) systém. Systém

dokázal synchronizovať dvoch robotov na jednom pracovisku. MRC tiež umožnilo kontrolovať beh programu robotov pomocou obyčajného počítača. Okrem sledovania však umožnil aj úpravy vykonávaného programu a nahratie úprav späť do kontroléru robota. O dva roky neskôr v roku 1996 spoločnosť KUKA vytvorila prvý kontrolér založený na platforme Windows. Inteligentný softvér nahradil obmedzujúci hardvér. Výrazne zjednodušil manipuláciu a flexibilitu robotov. Podľa zdroja [5] tento koncept vyžadoval o 35% menej hardvérových súčiastok a 50% menej konektorov a káblov. V roku 2006 sa objavu bezdrôtový ovládací panel WiTP (Wireless Touch Pendant). Komunikácia medzi robotom a ovládacím panelom prebiehala bezdrôtov. Operátor mohol ovládať viacero jednotiek pomocou jedného ovládacieho prvku a bez nutnosti presunu medzi robotickými jednotkami.

### 2.3.3 Súčasnosť a budúcnosť

Koncom deväťdesiatych rokov začali pribúdať nové technológie umožňujúce identifikáciu záťaže pre zlepšovanie výkonu a detekciu kolízií. Tieto technológie viedli k zvyšovaniu bezpečnosti práce pri manipulácii s robotmi. V roku 2002 [18] sa objavuje prvý prípad priamej spolupráce medzi pracovníkom a robotom. Vývoj priemyselných robotov sa zameriava hlavne na zdokonalenie technológií. Zníženie váhy robotov, zvýšenie efektivity a rýchlosti práce. Prevratom sa stal v roku 2006 robot LBR 4 od spoločnosti KUKA. Telo robota bolo postavené z hliníku, čo znížilo váhu na 16kg. Robot bol vybavený integrovanými senzormi. Vďaka týmto senzorm dokázal robot monitorovať sily pôsobiace na kĺby robota. Ako jeden z prvých priemyselných robotov tento robot umožňoval prácu na zdieľanom pracovisku. Bližšie informácie o nasledujúcej verzii tohoto robota sú popísané v sekcii 2.4.3. Motoman v roku 2007 navrhol dizajn robota, pomocou ktorého urýchlil rýchlosť cyklu takmer o 15%. Spoločnosť FANUC v roku 2008 vytvorila robota so záťažovou váhou až 1200kg. Tento robot prekonal aj robota od spoločnosti KUKA, z roku 2007, s nosnou silou 1000kg.

V poslednom desaťročí sa objavuje nový pojem Industry 4.0 resp. Priemysel 4.0. Tento pojem vznikol ako výsledok iniciatívy Nemecka podporiť vývoj nových technológií v automatizačnom priemysle a inteligentnej domácnosti. Myšlienka Industry 4.0 je natoľko revolučná, že sa považuje za štvrtú priemyselnú revolúciu. Industry 4.0 má predstavovať kompletnú automatizáciu a robotizáciu veľkej väčšiny ľudských činností za účelom zvýšenia efektivity práce, využitia materiálu, zníženia ceny produktu a oslobodenia človeka od stereotypných činností.

V priemysle ide o vytvorenie tzv. „inteligentné továrne“ (Smart Factories), ktorých hlavná myšlienka je založená na pripojení všetkých zariadení do kyberneticko-fyzikálneho systému (Cyber-physical System – CPS), v ktorom všetky časti výrobného procesu (senzory, materiály, diely, stroje) spolu vzájomne komunikujú a navzájom sa ovplyvňujú. Použitím RFID (Radio Frequency Identification) čipov majú vzniknúť inteligentné produkty, ktoré majú presne zapísanú činnosť a vedú rozhodovať resp. reagovať v závislosti na meniacich sa požiadavkách zákazníka. V prípade zlyhania niektorého výrobného procesu vedú rozhodnúť o alternatívnej ceste k dosiahnutiu výsledného produktu. Industry 4.0 má ovplyvniť život človeka v mnohých oblastiach. Podrobný koncept je opísaný v [22].

Okrem vývoja v priemysle je kladený veľký dôraz na rozvoj sociálnych a vedeckých robotov. V roku 2003 boli na prieskum Marsu použité roboty Sprint a Opportunity. Ich cieľom bolo skúmanie povrchu a hľadanie známk vody na Marse. Použitie robotov na vesmírne misie pokračovalo v roku 2011 kedy NASA vypustila do vesmíru prvého Robonauta. Robonaut je humanoidný robot, ktorého cieľom je podieľať sa na vesmírnych misiách, spolupracovať s človekom a zároveň plniť úlohy, ktoré môžu byť príliš riskantné pre človeka [18].

## 2.4 Rozdelenie robotov

S nasadzovaním priemyselných robotov sa vyvíjali požiadavky na bezpečnosť pri práci. Ako prvé boli nasadzované klasické konvenčné roboty, ktoré postupom času možno nahradia nové kolaboratívne roboty. Táto sekcia je zameraná na popis rozdielov medzi konvenčnými a kolaboratívnymi robotmi, ale aj na prehľad a porovnanie rôznych kolaboratívnych robotov.

### 2.4.1 Konvenčné roboty

Prvé hydraulicky riadené roboty boli veľké a silné. Nasadzovanie konvenčných robotov zefektívnilo prácu, ale zvýšilo riziko úrazu na pracovisku. Z tohoto dôvodu bolo zavedené tzv. rozdelené pracovisko.

Roboty sú chránené bezpečnostnými kliečkami resp. zábranami. Takéto roboty sú zväčša používané na jedinú cyklickú akciu, ktorá nevyžaduje pozornosť pracovníka. Robot je počas celého chodu oddelený od pracovníkov a nemožno k nemu prísť. Tento typ robotov najčastejšie nachádza uplatnenie v činnostiach ako sú napr. zváranie, obrábanie, montáž a iné.

### 2.4.2 Kolaboratívne roboty

**Definícia 1.** *Cobot (Collaborative robot) je robot, pracujúci v priamom kontakte s človekom [15].*

Ako vyplýva z definície 1, oproti konvenčným robotom, kolaboratívne roboty pracujú bez ochranných kliečok, zábran na zdieľanom pracovisku s pracovníkom. Schopnosť bezpečne pracovať bez ochranných kliečok dodáva robotom väčšiu flexibilitu. Vďaka ich menšej veľkosti a nižšej hmotnosti je ich väčšinou jednoducho presúvať alebo meniť ich pracovnú činnosť. Bez nutnosti použitia ochrannej kliečky zaberajú na pracovisku omnoho menší priestor. Naopak oproti veľkým konvenčným robotom, kolaboratívne roboti nedosahujú rovnakú záťažovú nosnosť a majú presne stanovené maximálne pracovné rýchlosti. S tohoto dôvodu pracujú pomalšie ako bežné konvenčné roboty. Náročnosť programovania kolaboratívnych robotov závisí na type robota. Niektoré roboty možno programovať navádzaním robotických ramien, iné vyžadujú komplexné programy a presne naprogramované bezpečnostné podmienky.

Pri práci na zdieľanom pracovisku je kladený hlavný dôraz na bezpečnosť pri práci. Kolaboratívne roboty musia spĺňať prísne bezpečnostné kritéria. V roku 2013 Robotic Industries Association (RIA) vydala nový robotický štandard, ktorý pridáva nové bezpečnostné požiadavky pre roboty pracujúce na priemyselných pracoviskách bez bezpečnostných zábran. Dodnes nie sú vydané žiadne oficiálne normy pre prácu s kolaboratívnymi robotmi. Normy pre ich použitie sú rovnaké ako u ostatných priemyselných robotov. Požiadavky pre priamu spoluprácu sú popísané v normách [6] a [7].

Podľa uvedených ISO noriem, kolaboratívny robot musí spĺňať aspoň jednu z nasledujúcich vlastností:

**Monitorovanie okolia pre možnosť bezpečného zastavenia** je spôsob používaný v prípadoch, kedy robot pracuje nezávisle na človeku (prípadne v ochrannej kliečke). Pre zistenie prekážky v priestore robota sa používa jeden prípadne viac senzorov. Pri porušení bezpečnostných parametrov je chod robota pozastavený. Robot je spustený, sú aktivované brzdy. Po prijatí bezpečnostného signálu od pracovníka sa robot vracia ku svojej činnosti.

**Ručné navádzanie robota** za účelom naučenia novej dráhy. Senzory zaznamenávajú silu a krútiaci moment, ktorú vykoná pracovník pri navádzaní robota. Namerané hodnoty sú použité pri následnom učení robota. Tento spôsob používa napríklad kolaboratívny robot LBR iiwa.

**Monitorovanie zón** okolia robota je neustále monitorované pomocou laserov alebo iných obrazových technológií, ktoré zaznamenávajú pozíciu pracovníka. Pomocou týchto technológií je priestor v okolí robota delený do bezpečnostných zón. Ak pracovník alebo iná prekážka vstúpi do niektorej z preddefinovaných zón chodu robota, ten sa spomalí na rýchlosť, ktorá z pohľadu bezpečnosti a ochrany zdravia pri práci človeka je normou prípustná. Na rozdiel od metódy Monitorovania okolia pre bezpečné zastavenie, táto metóda nepožaduje žiadne signály pre opätovné uvedenie robota do činnosti. Chod robota je riadený bezpečnostnými zónami.

**Obmedzenie sily a výkonu** je najpoužívanejšia technológia v kolaboratívnej robotike. Technológia kombinuje niekoľko rôznych vlastností napr. monitorovanie okolia, použitie tenzometrických snímačov. Tieto roboty majú navrhnutý bezpečnostný dizajn, ktorý chráni používateľa pred zraneniami v dôsledku kolízie s robotom. Normy [6, 7] presne špecifikujú maximálnu silu a energiu, ktorá môže byť zaznamenaná snímačmi robota bez vzniku nebezpečenstva pri práci s robotom. Tieto roboty sú vhodné pre priame používanie s pracovníkom v rôznych úlohách.

### 2.4.3 Prehľad kolaboratívnych robotov

V sekcii sú popísané aktuálne vyvíjané kolaboratívne roboty, ich vlastnosti a možnosti nasadenia.

#### KUKA – LBR iiwa

Spoločnosť KUKA začala s vývojom LBR (Leichtbauroboter) resp. LWR (Lightweight Robot) v roku 1991, kedy vytvorila prvého robota LBR 1. Prvá predávaná verzia bola LBR 4 z roku 2006, ktorá sa stala základom pre súčasný model LBR 5 z roku 2013. Robot bol navrhnutý pre inteligentnú spoluprácu na priemyselnom pracovisku podľa čoho dostal názov iiwa (Intelligent Industrial Work Assistant). Dizajn robota imituje ľudské rameno, čiže vytvára otvorený kinematický mechanizmus so siedmimi rotačnými osami. V každom kĺbe má zabudované citlivé tenzometrické snímače, ktoré spolu s integrovaným softvérom zaručujú extrémnu citlivosť a bezpečnosť pri práci s človekom. Telo robota je vyrobené zo zliatiny hliníka a povrch robota nemá žiadne ostré hrany. Robot sa vyrába vo dvoch verziách :

- hmotnosť 23,9 kg, záťažová nosnosť 7 kg, dosah 800 mm
- hmotnosť 29,9 kg, záťažová nosnosť 14 kg, dosah 820 mm

Robota je možné navádzať a učiť ručne alebo pomocou KUKA smartPADu. Oproti verzii LBR 4, ktorá používala bežný kontrolér spoločnosti KUKA KRC, verzia LBR 5 je založená na novej architektúre kontroléru. Táto architektúra priniesla nové vývojové prostredie KUKA Sunrise.

Kontrolér umožňuje tri módy [21]:

- V móde „Position controller“ sa chová ako klasický konvenčný robot a pohybuje sa definovanou rýchlosťou po stanovenej dráhe.

- V móde „Gravity compensation“ dokáže kompenzovať hmotnosť nákladu a pohybovať sa pri vynaložení minimálnej vonkajšej sily.
- V móde „Compliance controller“ je možné riadiť rotačný a posuvný pohyb pre všetky osy robota. Je tiež možné definovať aktívnu silu pri pohybe robota.

Vďaka presnosti až do 0,1 mm je iiwa vhodná na montáž malých častí, presúvanie, testovanie a iné.



Obr. 2.1: Robot LBR iiwa od spoločnosti KUKA, prevzaté z [4].

### **Rethink Robotics – Baxter, Sawyer**

Baxter je prvý kolaboratívny dvoj-ramenový robot. Na trh bol uvedený v roku 2012 spoločnosťou Rethink Robotics. Konštrukcia robota umožňuje široký rozsah uplatnenia. Telo robota pozostáva z hlavy, trupu a dvoch ramien. Na hlave robota je upevnený tristošesťdesiat stupňový sonar a kamera pre detekciu ľudí v okolí robota. Ramená umožňujú sedem stupňov voľnosti. Je ich možné navádzať pomocou video-systému alebo manuálne pomocou užívateľského rozhrania. Kĺby robota obsahujú silové snímače pre zvýšenie bezpečnosti [24].

Robota je možné jednoducho učiť nové činnosti, nevyžaduje vnútorné programovanie. Obe ramená dokážu pracovať nezávisle na rôznych činnostiach alebo spolupracovať na jednej činnosti. Baxter zahŕňa bezpečnostné funkcie, ktoré mu umožňujú prehodnotiť cestu pohybu v prípade, keď dôjde k zmene jeho okolia. Touto vlastnosťou dokáže napodobiť správanie človeka na pracovisku a prispôbiť sa spoločnej spolupráci na zdieľanom pracovisku.

Sawyer je menší, ale poskytuje rovnaké bezpečnostné prvky ako Baxter. Jedná sa o jedno robotické rameno. Rameno má zabudovaný video systém, ktorý umožňuje detekciu prekážok pri uchopovaní predmetov. Sawyer používa inú technológiu pohonov. U Baxtera boli pružiny vyrobené z ocele, v tvare písmena „C“, zatiaľ čo Sawyer používa pružiny v tvare „S“, vyrobené z titánu [13]. Tento návrh dodáva robotickým kĺbom väčšiu tuhosť, čím zvyšuje presnosť pohybov. Sawyer nachádza uplatnenie pri montáži, obsluhu strojov, balení.

### **ABB – YuMi**

Kolaboratívny robot YuMi bol uvedený na trh v Apríli roku 2015, spoločnosťou ABB. Pôvodne bol robot nazvaný Frida (Friendly Robot for Industrial Dual-Arm), neskôr bol



Obr. 2.2: Robot Baxter, prevzaté z [3].

premenovaný na YuMi (for you and me). Jedná sa o dvojité robotické rameno, kde každé rameno dosahuje sedem stupňov voľnosti a záťažovú hmotnosť 0,5 kg. Svojou veľkosťou zaberá na pracovisku rovnaký priestor ako ľudský pracovník a vďaka váhe 38 kg je možné ho jednoducho presúvať medzi rôznymi pracoviskami.

Základom každého ramena je ľahká, ale pevná magnéziová kostra obalená mäkkou výplňou. Tento návrh umožňuje veľmi citlivú detekciu cudzieho predmetu. Pri kolízii s pracovníkom je robot schopný okamžite zastaviť svoj pohyb. Bezpečne navrhnutý povrchový dizajn prispieva k zvýšeniu bezpečnosti pri priamom kontakte s človekom.

YuMi bol navrhnutý na spájanie a prácu s malými, elektronickými súčiastkami. Ramená sú vybavené kamerovým systémom, pre detekciu predmetov a špeciálnym uchopovačom pre jednoduché uchopenie malého predmetu. Pri maximálnej rýchlosti 1500 mm/s dokáže robot pracovať s presnosťou 0,02 mm [13].



Obr. 2.3: Robot YuMi od spoločnosti ABB, prevzaté z [1].

## ABB – Roberta

V roku 2015 spoločnosť ABB odkúpila robota Roberta od spoločnosti Gomtec. Roberta je malý, flexibilný kolaboratívny robot. S váhou 19,5 kg dosahuje záťažovú hmotnosť 8 kg. Roberta má v každej zo šiestich osí zabudovaný bezpečnostný mechanizmus pre monitorovanie síl pôsobiacich na rameno robota. K Robetre je možné pripojiť bezpečnostný uchopovač, ktorý pomocou kamerového systému dokáže detekovať cudzie predmety. Koncový efektor má zabudované silové senzory, ktoré prispievajú k zvýšeniu bezpečnosti pri práci. Okrem bezpečnosti je možné, zabudovaný kamerový systém použiť pre detekciu predmetov, s ktorými robot pracuje [13].



Obr. 2.4: Robot Roberta, prevzaté z [8].

## Universal Robots – UR10, UR5, UR3

UR10, UR5 a UR3 je trojica ľahkých, flexibilných kolaboratívnych robotov. Roboty majú podobnú konštrukciu tela. Ide o jedno robotické rameno so šiestimi otočnými osami. Odlišujú sa veľkosťou a záťažovou hmotnosťou. V tabuľke 2.1 sú zhrnuté technické parametre všetkých verzií robotov.

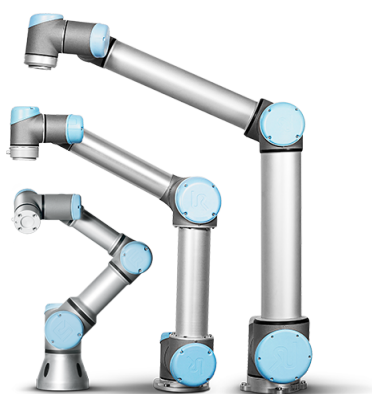
Roboty dosahujú presnosť do 0,1 mm. Sú vybavené bezpečnostným systémom založením na meraní síl v ramene robota. Bezpečnostný systém obsahuje módy na základe, ktorých robot reguluje svoju rýchlosť. V prípade prekročenia stanovených podmienok dokáže robot zastaviť svoju činnosť do 1250 ms [13].

Rada robotov UR umožňuje jednoduché programovanie pomocou učenia prechodových bodov, z ktorých je počítaná trajektória pohybu.

	UR10	UR5	UR3
Hmotnosť [kg]	28.9	18.4	11
Dosah [mm]	1300	850	500
Záťažová hmotnosť [kg]	10	5	3
Využitie	balenie montáž paletizácia	prekladanie testovanie	spájanie, lepenie maľovanie, prekladanie laboratórne práce

Tabuľka 2.1: Porovnanie rady robotov Universal Robots





Obr. 2.5: Roboti UR10, UR5 a UR3, prevzaté z [10].

### Fanuc – CR-35IA

CR-35IA je prvý kolaboratívny robot spoločnosti Fanuc. Činnosti ostatných kolaboratívnych robotov sú limitované nízkou záťažovou hmotnosťou. CR-35IA je prvý kolaboratívny robot, ktorý má záťažovú hmotnosť až 35kg a dokáže bezpečne pracovať na spoločnom pracovisku bez ochranných kliebok. Aby bolo možné kolaboratívneho robota jednoducho odlíšiť od bežných priemyselných robotov, má výraznú zelenú farbu. Robot bol primárne navrhnutý na cyklické činnosti a kvôli jeho vysokej hmotnosti nieje odporúčaná zmena pracoviska. Základ tela je robot M-20iA/35M. Tento robot je vložený do gumenej kostry, ktorá zakrýva ostré hrany robota. Pomocou silových senzorov a upraveného softvéru robot dokáže zaznamenať kontakt s človekom a bezpečne zastaviť. Fanuc k robotovi poskytuje videosystém alebo 3D systém, ktorý umožňuje detekciu predmetov.

Ako ostatné kolaboratívne roboty aj CR-35IA je možné jednoducho navádzať. Robot je vhodný na zdvíhanie ťažkých predmetov, paletizáciu, montáž a iné [26].



Obr. 2.6: Robot CR-35IA spoločnosti Fanuc, prevzaté z [9].

## KAWADA INDUSTRIES – NEXTAGE

V roku 2015 vydala kolaboratívneho robota aj Japonská spoločnosť Kawada Industries. Robot je považovaný za ďalšiu generáciu priemyselnej robotiky, podľa čoho dostal názov NEXTAGE (Next Generation Industrial Robot). Aj napriek tomu, že sa nejedná o humanoidného robota telo má zložené z hlavy, trupu a dvoch ramien. Dokopy dosahuje až 15 stupňov voľnosti. Šesť stupňov v každom remene, dva stupne pre rotáciu hlavy a jeden v trupe. Všetky osy pracujú na nízko príkonovej regulácii do 80 W. Roboty s príkon nižším ako 80 W môžu byť voľne použité bez ochranných klieťok.

Na hlave má robot umiestnenú stereoskopickú kameru, ktorá umožňuje diagnostiku okolia robota. Vďaka tejto kamere sa dokáže robot jednoducho adaptovať na zmeny prostredia. V prípade zmeny pozície dokáže upraviť výchylky spôsobené posunutím. Kamery sú umiestnené aj vedľa koncových efektoroch oboch ramien. Tieto kamery slúžia k rozpoznávaniu tvaru a vzdialenosti predmetu. Unikátna je špeciálne navrhnutú štruktúru ramien, ktorá nedovoľuje pohyb mimo robotické okolie. Táto konštrukcia zabraňuje nežiadúcemu stretu s pracovníkom. Na trupe robota je umiestnený displej, ktorý zobrazuje stavové informácie robota [11].

Robot má navrhnutý vlastný softvér, ktorý umožňuje programovanie v jazyku .net. Robota je možné navádzať aj pomocou integrovaného grafického prostredia.

Okrem robota NEXTAGE, Kawada Industries, poskytuje aj robota HIRO. HIRO je prispôbená verzia robota NEXTAGE. Softvér tohoto robota spadá pod GNU (General Public License), čo poskytuje lepšie možnosti pre vývoj robota.



Obr. 2.7: Robot Nextage spoločnosti Kawada Industries, prevzaté z [11].

### Zhrnutie

V súčasnosti prebieha masívny rozvoj týchto robotov. Mnohé spoločnosti vyvíjajú vlastný hardvér a softvér pri vytvorení ideálneho ľudského spolupracovníka. Pretože Matador Industries vyvíja aplikácie pre kolaboratívneho robota LBR iiwa, bol práve preto tento robot zvolený pre implementáciu aplikácie. Rovnako ako LBR iiwa, všetky spomenuté roboty dokážu pracovať vedľa pracovníka bez ochranných zábran. U rady z nich sa vyskytuje možnosť učenia pomocou navádzania, čo zvyšuje flexibilitu robota, a tým jeho znovu-použitelnosť na rôznych pracoviskách. Z výnimkou robota CR-35IA, spoločnosti Fanuc, sú všetky roboty zamerané prevažne na prácu s ľahkými predmetmi, prekladanie, montáž alebo testovanie.

## Kapitola 3

# Špecifikácia robota a vývojového prostredia

Praktická časť tejto práce je zameraná na vytvorenie aplikácie pre robot LBR iiwa. V sekcii [2.4.3](#) bola popísaná základná technická špecifikácia tohoto robota. Táto kapitola sa venuje podrobnejšiemu popisu jednotlivých častí robota, vývojovému prostrediu a implementačnému jazyku.

### 3.1 Robot LBR iiwa

Celý robot pozostáva z niekoľkých častí: z robotického ramena, KUKA Sunrise Cabinetu a KUKA SmartPAD-u. Najdôležitejšou časťou je KUKA Sunrise Cabinet, ktorý je priamo prepojený s robotickým ramenom a KUKA SmartPAD-om. KUKA Sunrise Cabinet sa stará o správu hardvéru a prepojuje bezpečnostnú, logickú, robotickú a procesnú jednotku.

Softvér riadi KUKA Sunrise OS, ktorý bol navrhnutý aby oddelil prácu operátora s robotom od programátorskej časti. KUKA SmartPad slúži na ovládanie robota, spúšťanie a ukončenie aplikácií, ale neumožňuje priame programovanie robota. Na vývoj nových aplikácií KUKA vyvinula prostredie Sunrise Workbench, ktoré typicky beží na externom počítači prepojeným s robotom. Poslednou softvérovou časťou je WorkVisual pomocou, ktorého možno konfigurovať zbernice.

### 3.2 Sunrise Workbench

Sunrise Workbench je vývojové prostredie vychádzajúce z prostredia Eclipse. Oproti Eclipse, Sunrise pridáva programovaciu perspektívu na programovanie robota. Prostredie je presne navrhnuté na tvorbu, inštaláciu a synchronizáciu robotických projektov. Umožňuje konfigurovať nastavenia pripojených nástrojov, prechodových bodov a tiež spravovať bezpečnostnú konfiguráciu robota.

Hlavným programovacím jazykom je Java. Tento jazyk pridáva možnosť objektovo-orientovaného prístupu, uľahčuje tvorbu aplikácie, kontrolu chýb pri vývoji aplikácie, tiež umožňuje rozširovať základnú funkcionálnosť, použitím veľkého množstva knižníc, ktoré sú v súčasnej dobe vytvorené pre jazyk Java. Pre ovládanie robota KUKA vytvorila robotickú knižnicu. Použitím tejto knižnice sa vytvárajú aplikácie a konfigurujú všetky nastavenia robota.

## 3.3 Konfiguračné nastavenia robota

Robot poskytuje veľké množstvo konfiguračných nastavení, pomocou ktorých je možné programovať správanie robota. V tejto sekcii sú uvedené niektoré nastavenia, s ktorými bolo pracované pri implementácii robota.

### 3.3.1 Súradný systém

Robot podporuje niekoľko súradných systémov. Jednotlivé systémy možno nastaviť pomocou KUKA SmartPadu-u. Hlavný a prednastavený súradný systém je Svet (World). Jedná sa o karteziánsky súradný systém, ktorý je základom pre všetky ostatné koordinačné systémy, ktoré podporuje robot. Počiatok súradného systému je definovaný v základni robota. Aby robot mohol dosiahnuť požadovaný bod v priestore má každý bod definovaných šesť súradníc. Vzdialenosti X,Y,Z sú počítané od počiatku súradnicovej sústavy a slúžia pre definíciu pohybu po osiach X,Y,Z. Uhly A,B,C pre otočenia v osiach X,Y,Z.

### 3.3.2 Preddefinované frame-i

Pre definovanie bodov v priestore robot využíva tzv. frames. Frame je troj-dimenzionálny súradný systém, ktorý je definovaný bodmi a relatívnou pozíciou k nadradenému súradnému systému. Pomocou frame-ou sú definované prechodové body v priestore, ktoré sú spolu s frame-ami uložené v stromovej štruktúre. Body a frame-i je možné vytvárať vo vývojovom prostredí a upravovať pomocou SmartPadu. Fyzicky sú frame-i uložené v xml súbore, z ktorého sú načítané pri pohybe robota.

### 3.3.3 Preddefinované nástroje

Pre dosiahnutie maximálnej presnosti pri práci robot požaduje definovať nástroje, ktoré sú pripevnené na konci ramena. Nástroj má definovanú hmotnosť a ťažisko. Každému nástroju je možné priradiť TCP body. TCP (Tool Center Point) body sú frame-i, ktoré popisujú pracovné body robota. Každý TCP bod je definovaný relatívnou pozíciou oproti nadradenému frame-u.

### 3.3.4 Monitorovacie mechanizmy

Bezpečnostný koncept Sunrise definuje dva monitorovacie mechanizmy: PSM (Permanent Safety Monitoring) a ESM (Event-driven Safety Monitoring). Mechanizmus PSM zachováva aktívne všetky bezpečnostné funkcie.

ESM state dovoľuje definovať vlastné bezpečnostné stavy a prepínať medzi nimi počas behu robotického aplikácie. Stavom môžu byť definované vlastné pohybové parametre napríklad maximálna rýchlosť alebo podmienky prerušenia pohybu pri kolízii. Tento mechanizmus možno využiť pri priamej spolupráci človeka s robotom.

### 3.3.5 Pohyby

Všetky pohyby robota sú vykonávané vzhľadom na preddefinovaný TCP bod pripevneného nástroja. Robot pozná 4 druhy pohybov: **PTP** (Point To Point), **LIN** (Linear), **CIRC** (Circular) a **SPL** (Polynomial). Jednotlivé druhy pohybov môžu byť rôzne kombinované a spájané do pohybových sekvencií, ktoré tvoria výsledný pohyb.

**PTP** je kľbový pohyb robota, pri ktorom sa robot pohybuje najrýchlejšou možnou cestou do cieľového bodu. Pri pohybe robota je dôležité natočenie jednotlivých osí robota, z tohoto dôvodu nemusí byť vždy najkratší pohyb najrýchlejší. Pri PTP pohybe nemožno predom určiť presnú trajektóriu pohybu.

**LIN** je pohyb, pri ktorom sa robot pohybuje po úsečke tvorenej počiatočným a koncovým bodom pohybu.

**CIRC** je kruhový typ pohybu, ktorý je definovaný troma bodmi: počiatočným, koncovým a pomocným bodom.

**SPL** umožňuje vytvárať plynulé pohyby po krivkách. SPL pohyby sú združované do blokov. Bloky združujú jednotlivé pohyby a sú chápané kontrolérom ako jeden plynulý pohyb.

### 3.3.6 Zaznamenávanie dát

Snímače krútiaceho momentu v každom kľbe robota dokážu zaznamenávať sily pôsobiace na osi počas pohybu. Pomocou robotickej knižnice je možné získať silu pôsobiacu v každej osi robota. Sily a krútiace momenty môžu byť ukladané do súboru, a v prípade potreby neskôr príslušne spracované.

### 3.3.7 Pohybové podmienky

Pri pohybe je možné definovať stavové podmienky, ktoré zaistia zastavenie, spustenie robota alebo inú akciu. Systém overuje aktuálne namerané hodnoty so stanovenými limitmi. V prípade, kedy je niektorý limit prekročený je vyvolaná príslušná akcia. Robot umožňuje pracovať s veľkým množstvom podmienok ako napríklad: silová podmienka (prekročenie limitu sily pôsobiacej v osiach v X,Y,Z), podmienka booleanových signálov (kontroluje jeden vstupný bit, vyvolaná akciu podľa stavu bitu), podmienky zastavenia pohybu (podmienka ovplyvňujúca pohyb robota) a rôzne iné podmienky.

## 3.4 Komunikácia s okolím

Pre vytvorenie komunikácie s vonkajším prostredím musí mať robotický kontrolér natavenú vonkajšiu IP adresu. Robot má definovaný rozsah adries, ktoré využíva pre vnútornú konfiguráciu. Podobne je to s rozsahom portov povolených pre TCP alebo UDP komunikáciu. Povolené výstupné porty sú v rozsahu 30000 – 30010.

## Kapitola 4

# Analýza a návrh práce

Praktická časť aplikácie pozostáva z dvoch častí: robotickej a desktopovej aplikácie, ktoré musia navzájom komunikovať. V tejto kapitole je popísaná analýza problému, podrobný návrh oboch aplikácií, spôsob výmeny a uloženia dát.

### 4.1 Analýza robotickej aplikácie

Pri analýze robotickej aplikácie som uvažovala o dvoch spôsoboch implementácie.

- cyklické testovanie jediného tlačidla,
- testovanie viacerých tlačidiel v presne definovanom podarí.

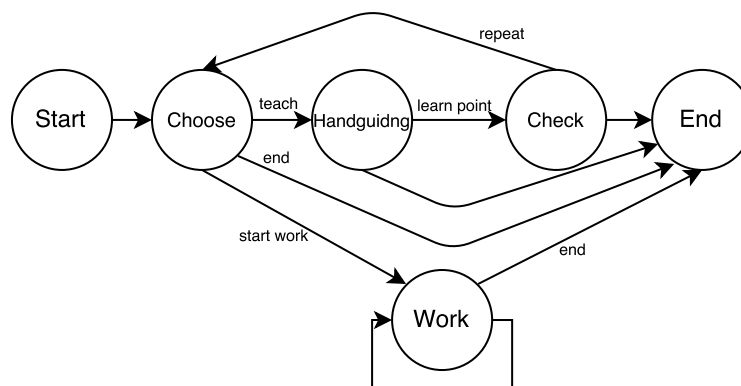
Pri cyklickom testovaní jedného tlačidla robot opakuje rovnakú akciu, pokiaľ pracovník neukončí jeho činnosť. Dôležitou časťou návrhu je stanoviť spôsob, ktorým robot dostane informáciu o polohe, v ktorej sa nachádza testovaná súčiastka. Pre presné dosiahnutie požadovanej pozície musí mať robot definovaný nástroj, ktorý je upevnený na jeho konci, a bod TCP (Tool Center Point), podľa ktorého počíta pozíciu pripevneného nástroja. Následne je možné robota definovať ostatné body v priestore, cez ktoré sa vykonaná pohyb.

Prechodové body môžu byť nastavené priamo v programe SunriseWorkbench alebo pomocou pripojeného pripojeného smartPADu. Manuálny zápis bodov pomocou Sunrise Workbench je v praxi nepoužiteľný, pretože pracovník nemá prístup k vývojovému prostrediu ani programovej časti aplikácie. Posledné verzie robota LBR iiwa podporujú ručné navádzanie robota, ktoré sa dá využiť pri učení prechodových bodov. Vytvorením aplikácie je možné zapnúť navádzací mód robota a naučiť robot nové body napríklad pomocou integrovaného tlačidla na tele robota.

### 4.2 Návrh robotickej aplikácie

Po spustení aplikácie robot umožní pracovníkovi zvoliť úpravu bodu testovaného tlačidla. Činnosť robota je možné zobrazit pomocou jednoduchého, stavového automatu, ktorý je zobrazený na obrázku 4.1. V stave **Choose** je robot v počiatočnej pozícii Home. V prípade, keď užívateľ zvolí možnosť úpravy pozície bodu aplikácia prejde do stavu **Handguiding**. V tomto stave je možné robota ručne priviesť do požadovanej pozície a stanoviť nový prechodový bod. Po ukončení ručného navádzania je nutné skontrolovať správnosť naučeného bodu. Z tohoto dôvodu som pridala stav **Check**, ktorý vykoná prvý cyklus. Po prvom cykle

je činnosť robota vrátená do stavu **Choose**. V prípade, kedy pracovník nevyžaduje učenie nového bodu, aplikácia prechádza do stavu **Work**. V tomto stave robot cyklicky prevádza testovanie tlačidla, dokým nie je jeho činnosť prerušená pracovníkom.



Obr. 4.1: Stavový automat popisujúci činnosť robota.

Počas testovania robot meria sily pôsobiace na bod TCP. Sily môžu byť odosielané okamžite v krátkych časových intervaloch, napr. každé 3 ms, alebo spájané do jedného reťazca, ktorý bude odoslaný po dokončení pohybu. Vhodné riešenie bude zistené pri implementácii problému. Počas celého pohybu, robot kontroluje sily vo všetkých osiach. Ak dôjde k porušeniu stanovených bezpečnostných podmienok robot svoj pohyb zastaví a vráti sa do posledného bodu prechodu. Tento pohyb sa pokúsi následne zopakovať. Činnosť robota je zastavená ukončením aplikácie prostredníctvom KUKA smartPAD-u.

Návrh testovania viacerých tlačidiel počas jedného cyklu je mierne zložitejší ako testovanie jedného tlačidla. Najdôležitejšou časťou je určenie počtu tlačidiel na testovanom zariadení. Pri návrhu treba zväžiť rôznorodosť testovaných zariadení. Každá súčiastka môže obsahovať rôzny počet tlačidiel.

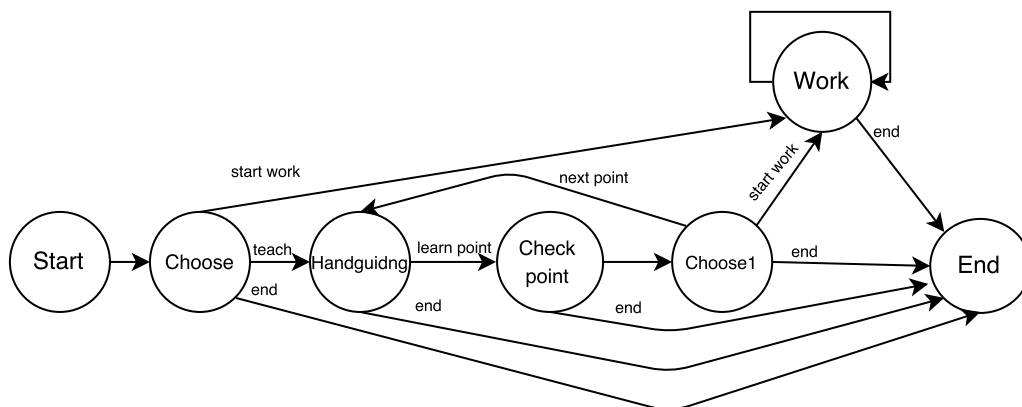
Stavový automat tohoto návrhu je zobrazený na obrázku 4.2. Stav **Choose** a **Handguiding** majú rovnakú funkčnosť ako v predošlom prípade. Stav **Check point** opäť vykoná jeden cyklus stlačenia a zároveň pridá možnosť rozhodnutia o uložení alebo zahodení naučeného bodu. V nasledujúcom stave **Choose1** užívateľ rozhodne či robot pridá ďalší bod (v tomto prípade sa robot vracia do stavu **Handguiding**) alebo zvolí možnosť začať v práci (presun do stavu **Work**). Učenie bodov je možné až dokým pracovník nezadá všetky požadované body. V stave **Work** robot prechádza všetky naučené body a meria sily v bode TCP. O každom bode je nutné uchovávať identifikátor, ktorý presne označuje testované tlačidlo. S týmto riešením musí byť tiež implementovaná desktopová aplikácia.

Rovnako ako pri prvom návrhu musí robot sledovať sily vo všetkých svojich osiach, aby sa predišlo zraneniam na pracovisku.

### 4.3 Analýza desktopovej aplikácie

Cieľom aplikácie je vytvoriť komplexný počítačový program s vlastným grafickým rozhraním a špecifickou funkčnosťou. Z tohoto dôvodu je návrh potrebné rozdeliť na návrh grafického užívateľského rozhrania (GUI) a návrh pozadia (*backend-u*).

Desktopová aplikácia je prepojená s robotom, podľa čoho je možné predpokladať, že cieľový užívateľ bude obsluhovať zároveň robot aj desktopovú aplikáciu. Cieľovou skupinou



Obr. 4.2: Stavový automat popisujúci činnosť robota.

užívateľov sú priemerne počítačovo-zdatný užívatelia. Užívateľ by mal byť schopný bez predošlého zaškolenia desktopovú aplikáciu používať. Z uvedeného dôvodu musí byť navrhnuté užívateľské rozhranie zrozumiteľné a intuitívne. Platforma pre beh aplikácie je Windows 7 a vyššia.

Robotická časť aplikácie požaduje implementáciu v jazyky Java. Pre jednoduchosť implementačného jazyka je vhodné zvoliť rovnaký jazyk aj pre implementáciu desktopovej časti. Jazyk Java navyše poskytuje rozšírenie JFC Swing, ktoré umožňuje tvorbu grafického rozhrania. Použitím tohoto jazyka je možné implementovať celú funkcionálnosť bez nutnosti prepájania jednotlivých častí v rôznych implementačných jazykoch.

*Backend* aplikácie požaduje pomerne rozsiahli návrh. Je treba navrhnuť spôsob prijímania, ukladania a zobrazovania dát. Najvhodnejší spôsob uloženia veľkého množstva dát je prostredníctvom databázy. Na uloženie dát postačí niektorý z voľne prístupných databázových serverov. Spoločnosť Microsoft poskytuje SQL Server Express, ktorý poskytuje 10 GB úložný priestor pre každú vytvorenú databázu. Spolu s nástrojom na správu SQL Serveru, Microsoft ponúka širokú sadu nástrojov pre prácu alebo zálohovanie databáz. Pre prácu v jazyku java poskytuje SQL Sever vlastnú knižnicu.

Inou variantov voľne prístupného databázového systému je PostgreSQL. PostgreSQL je voľne prístupný databázový systém, s neustálou podporou a vývojom. Systém je vhodný pre takmer všetky platformy Windows, UNIX a LINUX. Veľkosť databázy je neobmedzená, veľkosť jednej tabuľky je až 32 TB. PostgreSQL podporuje veľkú časť SQL štandardu zahŕňajúc dotazy, cudzie kľúče, trigger a iné. Podobne ako SQL Server poskytuje rozsiahle množstvo knižníc pre rôzne programovacie jazyky. Databázový server bude zvolený pri implementácii.

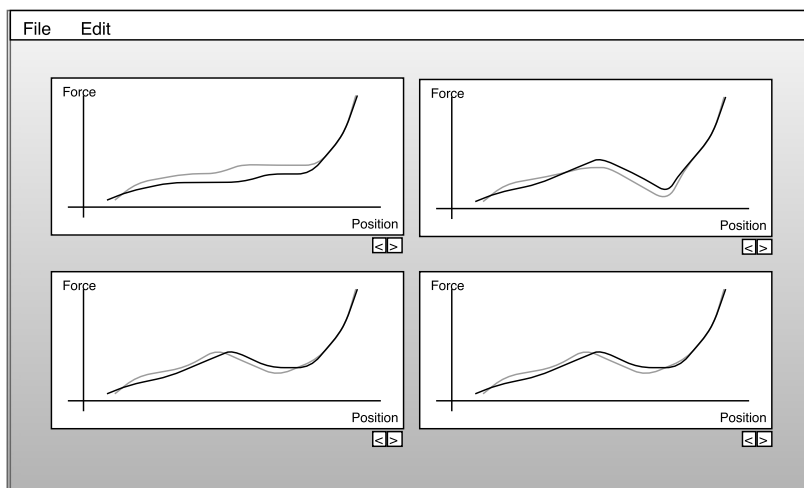
Zobrazenie nameraných dát je najdôležitejšou časťou desktopovej aplikácie. Pre vykreslenie grafu je možné použiť niektorú z existujúcich knižníc pre jazyk Java. Jednou z takýchto knižníc je JFreeChart. Táto knižnica umožňuje vykreslenie rôznych druhov grafov v profesionálnej kvalite. Knižnica poskytuje rozsiahlu funkcionálnosť, vrátane exportu grafov do rôznych výstupných súborov.



## 4.4 Návrh desktopovej aplikácie

Centrom grafickej plochy je graf, ktorý zobrazuje dáta prijaté od robota počas jedného stlačenia a uvoľnenia tlačidla. Je nutné uvážiť možnosť, kedy aplikácia musí zobrazovať súčasne dáta rôznych tlačidiel. Grafy tlačidiel je možné zobraziť vedľa seba alebo za sebou, kedy je užívateľ schopný vybrať graf, ktorý je zobrazený na popredí. Varianty sú zobrazené na obrázku 4.3 a 4.4. Prvý obrázok 4.3 zobrazuje rozloženie grafov, ktoré sú usporiadané po celej pracovnej ploche. Na tomto rozhraní vidí užívateľ všetky grafy súčasne, čo môže byť výhodou. Jeho nevýhodou je maximálny počet tlačidiel, ktorý musí byť známy pri tvorbe aplikácie. Z tohoto dôvodu je tento spôsob nie príliš vhodný. Na obrázku 4.4 je centrom obrazovky jeden graf ostatné grafy je možné zobraziť prepnutím pomocou záložiek. Tento spôsob umožňuje pridať väčší počet tlačidiel. Graf je väčší a užívateľ dokáže ľahšie rozoznať namerané hodnoty na grafe.

Na pracovnej ploche budú umiestnené tlačidlá, pomocou ktorých je možné prepínať aktuálny graf. Stlačením tlačidla budú zobrazované historicky namerané hodnoty. Export grafov je možný pomocou tlačidla alebo horného menu. Horné menu poskytuje možnosť úpravy, exportu grafu, uloženia, ukončenia aplikácie, zobrazenia dialógov. Dialógy slúžia na nastavenie pripojenia do databázy a zmeny serveru.

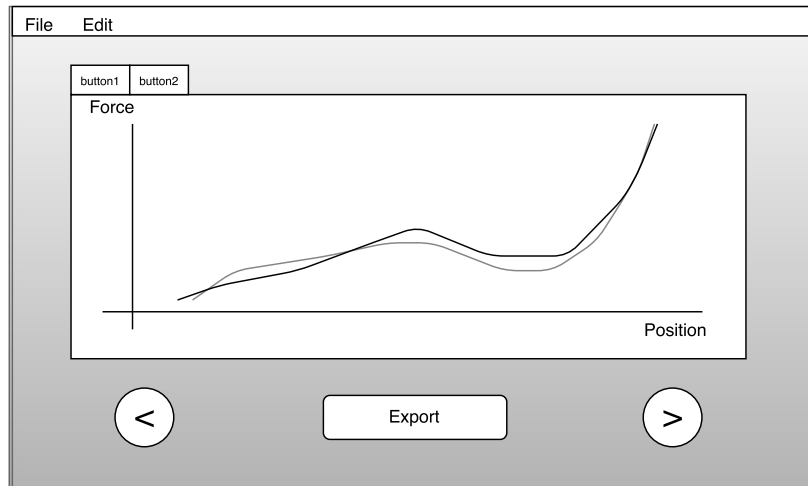


Obr. 4.3: Prvý návrh rozhrania desktopovej aplikácie.

Prijímanie dát bude zabezpečovať serverová časť aplikácie. Spojenie musí byť vytvorené hneď po spustení aplikácie. Server prijme správy obsahujúce dáta o stlačení a uvoľnení testovaného tlačidla. Dáta sú po prijatí uložené, následne môže prebehnúť vykreslenie grafu.

Uloženie dát môže byť navrhnuté nasledovne: pre každé testované tlačidlo je vytvorená jedna tabuľka databázy. Formát tabuľky je zobrazený na obrázku 4.5. Jeden riadok tabuľky značí jedno meranie tlačidla (stlačenie a uvoľnenie). Primárny kľúč tabuľky ID pridružuje číslo merania. `Serie_1`, `serie_2` ukladajú dáta, ktoré sú prijímané od robota a vykreslené do grafu. `Part_num` definuje číslo súčiastky. Užívateľ môže toto číslo manuálne nastaviť v aplikácii. Špecifické číslo súčiastky je potrebné v prípade, kedy je testované rovnaké tlačidlo na rôznych súčiastkach.

Vykreslený graf sa zmení s prijatím nových dát alebo po kliknutí na tlačidlá. Pri vykreslení sú na obrazovku vypisované informácie o aktuálne testovanom tlačidle. V prípade, keď dôjde ku chybe vytvorenia serveru, pripojenia do databázy, vykreslenia grafu alebo



Obr. 4.4: Druhý návrh rozhrania desktopovej aplikácie.

ButtonName	
PK	ID
	serie_1
	serie_2
	part_num
	date_time

Obr. 4.5: Vzor databázovej tabuľky pre uloženie údajov o jednom tlačidle.

inej chybe, program vypíše na obrazovku chybové hlásenie. Po ukončení aplikácie a opätovnom spustení musí aplikácia zachovať posledné nastavenia. Všetky údaje je možné ukladať do konfiguračných súborov. V prvom súbore budú uložené informácie o použítom porte pre server a údaje o databázovom serveri. V druhom súbore budú, vo formáte xml, uložené podrobné údaje o aktuálne zobrazených grafoch. Pre vytváranie a spracovanie xml je možné použiť knižnicu pre prácu s xml súborami.

## 4.5 Návrh komunikácie

Komunikácia medzi robotom a desktopovou aplikáciou má fungovať na protokole TCP. Táto architektúra definuje minimálne dvoch účastníkov: klienta a server. V súčasnej situácii je veľmi nepravdepodobné, že by desktopová aplikácia zbierala údaje naraz z rôznych zdrojov. Napriek tomu som sa rozhodla implementovať server na strane desktopovej aplikácie. Táto implementácia uľahčí a urýchli výkon robotической aplikácii. Na overenie spojenie je možné na začiatku robotической aplikácie poslať inicializačný paket. V prípade kedy odosielanie dát na server zlyhá, aplikácia vypíše chybové hlásenie na obrazovku KUKA smartPAD-u.

# Kapitola 5

## Implementácia

Nasledujúca kapitola vysvetľuje zvolený návrh riešenia a popisuje jednotlivé časti implementácie robotickej a desktopovej časti aplikácie.

### 5.1 Robotická aplikácia

Pre implementáciu robotickej časti som zvolila návrh podľa automatu zobrazeného na obrázku 4.1. Po spustení aplikácie operátor naučí robota jeden bod a tento bude testovaný dokým operátor neukončí činnosť robota. Pri práci s robotom som sa snažila minimalizovať prácu, ktorú musí vykonať pracovník pre uvedenie robota do činnosti. Z tohoto dôvodu som zvolila tento návrh, pretože kladie menšie požiadavky na obsluhu a funkčnosť.

K pohybu robotická aplikácia využíva jeden frame, ktorý obsahuje dva prechodové body (HOME a PUSH). Inicializácia bodov v súradnom systéme robota je vytvorená pred prvým spustením aplikácie, pomocou KUKA SmartPAD-u. Pretože aplikácia pri spustení získava informácie o týchto bodoch, bez ich definície by nemohla pracovať a vypísali by chybu na obrazovku SmartPAD-u.

Robot pri behu opakuje tri druhy pohybov. Prvý je bezpečný pohyb, ktorým sa robot pohybuje do pozície HOME. Jedná sa o PTP (Point to Point) pohyb, ktorý má stanovenú silovú podmienku 15 N. Ak by sila pôsobiaca na robot v ľubovolnej ose prekročila silu 15 N, robot preruší pohyb a vráti sa späť a pokúsi sa pohyb zopakovať. Pohyb opakuje dokým nedosiahne cieľový bod. Ďalšie dva pohyby sú lineárne. Týmito pohybmi robot stláča a uvoľňuje testované tlačidlo. Pri stlačení má robot definovanú silovú podmienku v ose Z. V tejto osi robot vykonáva pohyb stlačenia a uvoľnenia tlačidla. Súradný systém robota bol popísaný v 3.3.1. Robot sa pokúsi posunúť o 3 cm po ose Z v momente, kedy sila prekročí 10 N robot sa vracia do pozície, z ktorej pohyb začal. Silu 10 N som určila na základe vlastností testovaného tlačidla. Táto sila sa ukázala ako dostačujúca na zaznamenania celého stlačenia tlačidla.

#### 5.1.1 Beh aplikácie

Hlavná trieda vytvorenej aplikácie rozširuje základnú triedu robotickej knižnice pre tvorbu nových aplikácií `RoboticsAPIApplication`. Trieda obsahuje 3 metódy `initialize()`, `run()` a `dispose()`. Metódy sú vykonávané automaticky vnútorným systémom robota. Metóda `initialize()` slúži na deklaráciu a inicializáciu robotických premenných. V metóde je inicializovaný TCP bod (s ktorým pracuje robot), prechodové body HOME a PUSH, kontrolné módy pre definovanie pohybových limitov robota. Prebieha vytvorenie triedy pre

klienta, a zároveň je poslaná prvá kontrolná správa desktopovej aplikácii, pre overenie platnosti spojenia. Aplikácia zároveň ponúkne operátorovi možnosť výberu tlačidla, ktoré bude testované.

Metóda `run()` slúži na programovanie robota. Obsah metódy prebehne práve jeden krát, preto pre cyklické opakovanie je potrebné vytvoriť v tejto metóde nekonečný cyklus. Na začiatku metódy je implementovaný príkaz pre presun robota do počiatočnej pozície HOME, následne je na obrazovke SmardPAD-u zobrazený dialóg. Užívateľ má možnosť začať prácu robota alebo prepnúť do navádzacieho režimu, v ktorom je robota možné učiť nový bod. V prípade manuálneho navádzania je robot prepnutý do módu `ESM(2)`, ktorý bol spomenutý v sekcii 3.3.4. Po navedení do požadovanej pozície je naučení nový bod stlačením zeleného tlačidla na remene robota. Toto robotické tlačidlo slúži primárne k spusteniu aplikácie v automatickom režime, preto pri implementácii učenia bodu bolo nutné prepísať funkcionality tohoto tlačidla. Po ukončení manuálneho navádzania je tlačidlu vrátená jeho pôvodná funkcionality. Keď má robot definovaný nový bod vykoná prvý cyklus. Robot stlačí a uvoľní tlačidlo. Sily pri stlačení a uvoľnení sú zaznamenávané do súboru každé tri milisekundy. Tento interval sa ukázal ako dostačujúci na zobrazenie presného grafu z nameraných hodnôt. Dáta sú zaznamenávané zvlášť pri stlačení a uvoľnení tlačidla. Po ukončení pohybov sú namerané sily vybrané zo súborov a zobrazené na obrazovku SmardPAD-u vo forme grafu. Graf je vytvorený použitím knižnice `JFreeChat`<sup>1</sup>. Graf slúži pre overenie merania v prípade kedy operátor nie je spokojný s výsledným grafom môže opakovať učenie bodu. Naučený bod je zapísaný do XML súboru, ktorý obsahuje rámce a body celej aplikácie. Zapísanie do XML súboru umožňuje použiť tento bod pri opätovnom spustení aplikácie bez nutnosti opätovného učenia. Pri vedení robota do testovacieho cyklu robot vykonáva rovnaký pohyb ako prvom testovacím cykle. Narozdiel od prvého cyklu z nameraných síl nie sú vytvorené dátové osy grafu, ale sú vytvorené správy, ktoré klient odosiela desktopovej aplikácii. Každá správa obsahuje identifikátor tlačidla a namerané sily počas stlačenia alebo uvoľnenia tlačidla. Tento cyklus sa opakuje dokým nieje činnosť robota zastavená operátorom. Aplikáciu je možné ukončiť pomocou KUKA SmartPAD-u. Pri ukončení robotického aplikácie je volaná metóda `dispose()`.

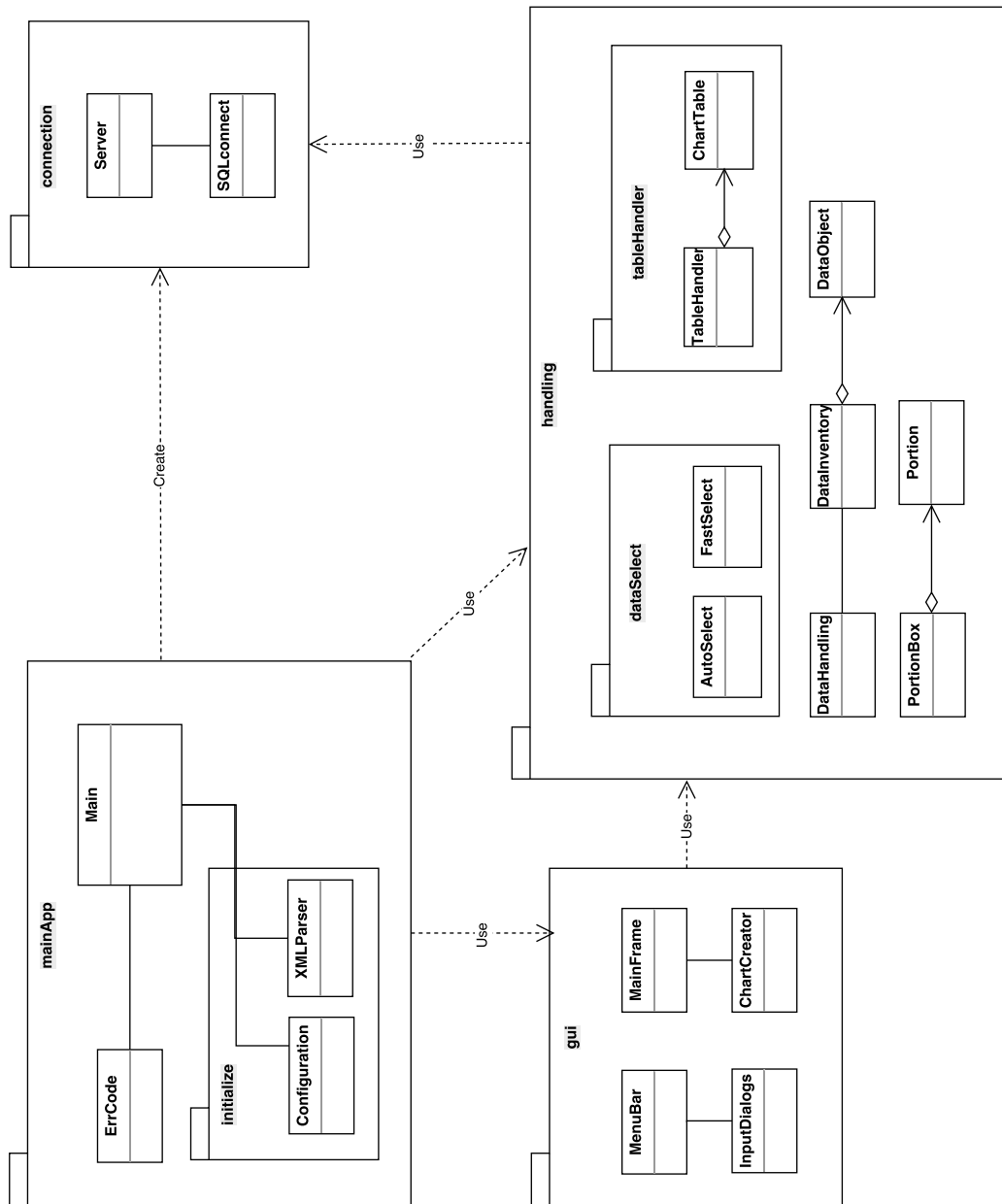
Pri implementácii sa ukázalo, že môže nastať chyba pri ukončení v prípade, keď operátor ukončí činnosť robota v momente, kedy sú údaje zapisované do súboru. Počas jedného zápisu je do súboru zapísaných približne 4000 hodnôt a proces zápisu nemôže byť prerušený. V prípade, že sa tak stane, aplikácia sa neukončí korektne. Tento stav však nespôsobí žiadny problém pri opätovnom spustení aplikácie.

## 5.2 Desktopová aplikácia

Pri výslednej implementácii som zvolila spôsob zobrazenia grafov pomocou záložiek. Grafy jednotlivých testovaných tlačidiel sú uložené za sebou. Užívateľ má možnosť zmeniť graf prostredníctvom záložiek umiestnených v ľavom hornom rohu grafu. Tento spôsob návrhu sa ukázal ako efektívnejší ale náročnejší na výslednú implementáciu. Implementácia jednotlivých častí je rozdelená v balíčkoch: hlavná časť (`MainApp`), grafické rozhranie (`GUI`), riadenie (`Handling`) a komunikácia (`Connection`). Diagram tried znázornený na obrázku 5.1 zobrazuje triedy v jednotlivých balíčkoch. Pre prehľadnosť nie sú v diagrame zobrazené vzťahy medzi jednotlivými triedami, závislosť naznačujú šípky medzi balíčkami. Bližšia funkcia balíčkov je popísaná v nasledujúcich sekciách.

---

<sup>1</sup>Viz <http://www.jfree.org/index.html>.



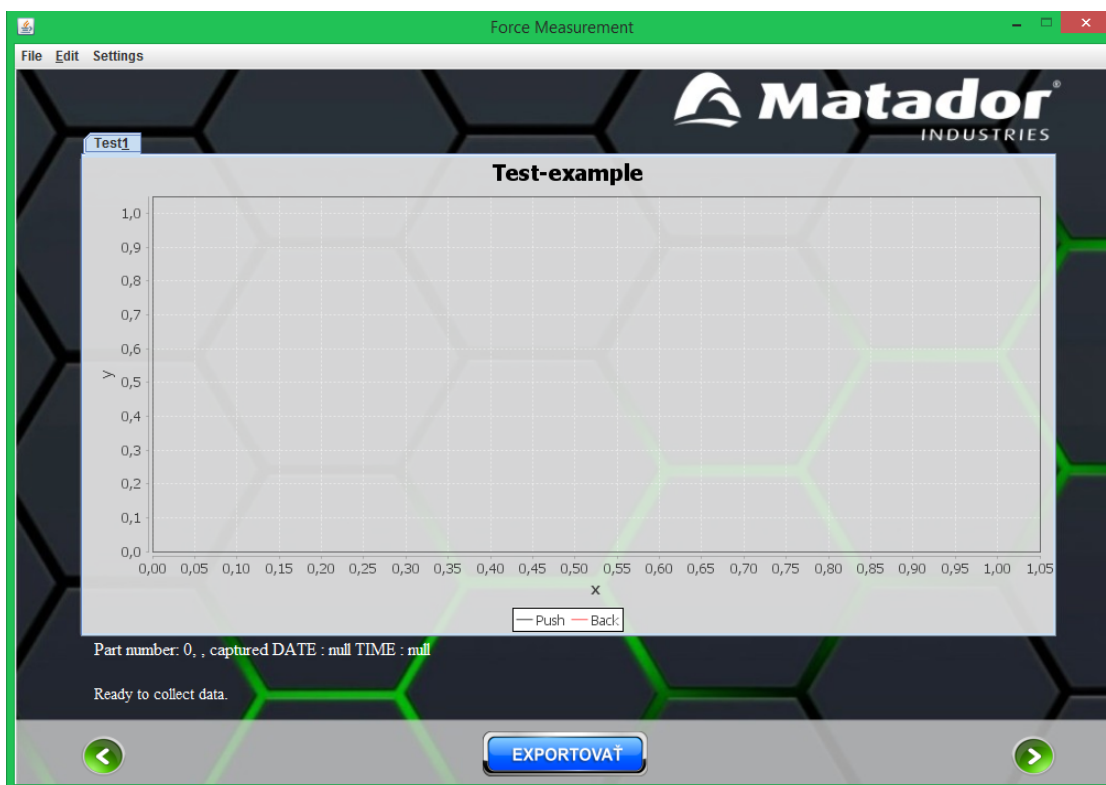
Obr. 5.1: Diagram tried desktopovej aplikácie.

### 5.2.1 Hlavná časť

Hlavná časť obsahuje triedy pre spustenie a inicializáciu aplikácie. Základom je trieda `Main`, ktorá vytvára základné okno pre grafické rozhranie, triedy pre inicializáciu, komunikáciu a riadenie aplikácie. Pri zavretí a opätovnom spustení aplikácie je potrebné aby sa aplikácia spustila v rovnakom stave ako bola ukončená. Je potrebné poznať počet testovaných tlačidiel a rôznych dielov, na ktorých boli testované tlačidlá. Taktiež je nutné rozpoznať informácie o testovanej databáze a porte na ktorom sa vytvára spojenie. O túto funkcionality sa starajú triedy `XMLParser` a `Configuration`. Informácie o testovaných tlačidlách a dieloch sú uložené v súbore `xmlFile`. Tento súbor je prepísaný vždy pri ukončení aplikácie. Pri spustení sú zo súboru načítané všetky informácie, podľa ktorých sú vytvorené záložky pre tlačidlá. Rovnako je to so súborom `config`, v ktorom sú informácie pre pripojenie do databázy.

### 5.2.2 Grafické rozhranie

Grafické rozhranie je zostavené pomocou JFC Swing. Plocha pozostáva z hlavného panelu, na ktorom sú umiestnené tlačidlá, menu a panel pre vykreslenie grafu dát. Dizajn jednotlivých grafických prvkov (tlačidlá, pozadie) bol poskytnutý spoločnosťou Matador v forme obrázkov. Obrázky boli priradené jednotlivým prvkom JFC Swing, čo prispelo k vytvoreniu výsledného grafického rozhrania. Vzhľad aplikácie bol vytvorený podľa návrhu na obrázku 4.4 a výsledné rozhranie je zobrazená na obrázku 5.2.



Obr. 5.2: Výsledné grafické rozhranie desktopovej aplikácie.

Položky menu sú vytvorené v triede `MenuBar` obsahujú možnosti pre pridanie nového testovacieho tlačidla, zmenu testovaného dielu, uloženie aktuálneho stavu, export zobraze-

ného grafu do formátu .pdf, a možnosť obnovenia pripojenia do databázy alebo serveru pre prijímanie dát. Pri požiadavke na pridanie nového testovacieho tlačidla je zobrazené dialógové okno. Do dialógového okna je možné vyplniť informácie o novej záložke (názov, popis) a vykreslenom grafe (názov grafu, osí). Po kliknutí na tlačidlo je **Finish** je vytvorená nová záložka s príslušným grafom. Číslo databázovej tabuľky, do ktorej budú ukladané dáta, je pridelený automaticky. Po vytvorení novej záložky je do menu v sekcii **Remove button** pridaná položka, ktorá umožňuje zmazanie vytvorenej záložky. Rovnakým spôsobom poskytuje menu možnosť pridania novej súčiastky. Po potvrdení dialógu program automaticky používa vytvorenú súčiastku. Pomocou menu je možné aktuálne testovanú súčiastku zmeniť. Všetky vstupné dialógy sú implementované v triede **InputDialog**s.

O vytvorenie grafu sa stará trieda **ChartCreator**. V tejto triede je pomocou knižnice **JFreeChart**<sup>2</sup> vytvorený nový panel, ktorý obsahuje výsledný graf. Metóde **createChart** je pomocou parametrov predaný názov grafu a osí. Trieda obsahuje tiež metódy pre vytvorenie zmenu a zmazanie dátových osí, ktoré tvoria výsledný graf. Dátové osy sú nastavené podľa dát nachádzajúcich sa v priradenej databázovej tabuľke.

### 5.2.3 Riadenie aplikácie

Riadenie aplikácie si vyžadovalo implementáciu pomerne náročného backgroundu. Na ploche môže byť súčasne niekoľko záložiek, ktoré obsahujú rôzne dáta. Každá záložka preberá dáta z inej databázovej tabuľky, zobrazuje rozličné meranie. Z tohoto dôvodu si aplikácia udržiava zoznam objektov, ktoré reprezentujú záložky na grafickej ploche. Aktuálne otvorené záložky sú uchované vo forme zrefazovaného zoznamu v triede **TableHandler**. Každý objekt v tomto zozname obsahuje údaje potrebné pre vykreslenie dát a to informácie o testovanom tlačidle, ktoré reprezentuje záložka, názov databázovej tabuľky v ktorej sú uložené dáta, informácie o grafe (názov grafu, osí), posledné meranie a aktuálne zobrazené meranie. Tieto informácie uchováva trieda **ChartTable**. Pri spustení aplikácie sú v každej záložke zobrazené grafy posledného merania uloženého v databázy. Pri stlačení tlačidla ľavej alebo pravej šípky je vykonaná obslužná udalosť nad záložkou, ktorá je aktuálne zobrazená v popredí. Udalosť vyvolaná po kliknutí na šípku zahŕňa vybratie správnych hodnôt z databázy a ich zobrazenie do grafu. Z databázy sú vždy vybrané dáta o jednom korektnom meraní v závislosti na poslednom zobrazenom meraní. To znamená pri stlačení ľavého tlačidla je načítané predošlé meranie pri stlačení pravého tlačidla nasledujúce meranie. V prípade, kedy sa v databázovej tabuľke žiadne ďalšie údaje nevyskytujú je na obrazovku vypísané príslušné hlásenie. Predpokladá sa, že dáta v databáze sú zoradené podľa času merania.

Počet hodnôt, z ktorých je zložený výsledný graf, sa môže v každom meraní líšiť, preto sú všetky dáta po vybratí z databázy dočasne uložené do zrefazovaného zoznamu, implementovaného v triede **DataInventory**. Následne sú prepísané dátové osy grafu, čím sa prekreslí zobrazený graf. Spolu s vykresleným grafom sú na obrazovku vypísané informácie o testovanom tlačidle, súčiastke, na ktorej sa tlačidlo nachádza, dátume, čase merania.

Zaujímavou funkcionalitou je zobrazovanie nameraných dát v reálnom čase. V prípade kedy sa po kliknutí na tlačidlo pravej šípky v databáze nenachádzajú žiadne ďalšie údaje aplikácia vytvorí nové vlákno (trieda **AutoSelect**), ktoré automaticky kontroluje posledné prijaté dáta. Pre aktuálne zobrazenú záložku sa nastaví premenná, ktorá signalizuje beh automatického režimu. V tomto režime sú novo prijaté dáta od robota ihneď zobrazené v grafe. Automatický režim je vypnutý pri kliknutí na tlačidlo ľavej šípky, v tomto prípade sú zároveň aj načítané dáta o predošlom meraní. Podobne ako pri vlákne v ktorom beží

<sup>2</sup>Viz <http://www.jfree.org/index.html>.

server, pri zatváraní aplikácie dôjde k vypnutiu všetkých automatických režimov, čo spôsobí ukončenie vlákien, následne nastane ukončenie celej aplikácie.

Súčasťou backgroundu aplikácie je aj exportovanie aktuálne zobrazených hodnôt. V aplikácii som implementovala export hodnôt do formátu PDF. Vo formáte PDF sú zobrazené aj informácie o aktuálne testovanej súčiastke, tlačidle a čase a dátume merania. Pre vytvorenie formátu PDF som použila knižnicu Itext<sup>3</sup>. Okrem exportu do spomínaných formátov sú pomocou knižnice JFreeChart pridané možnosti uloženia výsledného grafu do formátu svg alebo png.

#### 5.2.4 Komunikácia

Aplikácii nadväzuje dva druhy komunikácie: robotická - desktopová aplikácia a desktopová aplikácia - databáza. V komunikácii medzi robotom a počítačom plní desktopová časť rolu serveru. Pre implementáciu tejto triedy (TCPConnection) som použila BSD sockety, ktoré sú v jazyku Java dostupné prostredníctvom balíčku `java.net.*`. Server je vytváraný pri spustení aplikácie a po celú dobu chodu aplikácie beží v samostatnom vlákne, až dokým nieje jeho činnosť prerušená hlavnou aplikáciou. Spojenie sa vytvára na porte 30001, ktorý patrí do rozsahu povolených portov robota. V prípade, ak sa nepodarí vytvoriť spojenie na porte 30001 aplikácia zobrazí chybové hlásenie a spojenie je nutné reštartovať. Po úspešnom spustení server čaká dve správy od robotičkej aplikácie. Prvá prijatá správa požaduje dáta o stlačení tlačidla druhá o uvoľnení tlačidla. Dátam je pridelený čas prijatia, následné sú upravené do požadovaného formátu a uložené do prislúchajúcej databázovej tabuľky. Na všetky prijaté správy server odpovedá. Prijaté dáta sú ukladané do databázy aj v prípade, kedy v aplikácii neexistuje záložka, ktorá by tieto dáta zobrazovala. V situácii kedy sú dáta v neplatnom formáte alebo požadovaná databázová tabuľka neexistuje je vypísaná chybová hláška a všetky hodnoty sú zahodené. Pretože server beží v paralelnom vlákne bolo dôležité postarať sa o korektné ukončenie tohoto vlákna. Pri ukončení aplikácie sa hlavné vlákno pokúsi ukončiť server. Ten sa po obdržaní príkazu o ukončenie korektne ukončí, následne sa ukončí beh celej aplikácie.

Pre ukladanie dát som zvolila SQL Server<sup>4</sup> od spoločnosti Microsoft. Tento druh serveru poskytuje dostačujúce množstvo priestoru pre prijaté dáta a zároveň vhodné prostredie pre správu vytvorenej databázy. Pri implementácii som použila balíček JDBC, ktorý obsahuje základné triedy pre vytvorenie a komunikáciu s databázou. Spojenie na databázu je vytvorené z údajov, ktoré sú uložené v súbore config. Implementovaná trieda pre komunikáciu s databázou obsahuje metódy, ktoré slúžia na vytvorenie spojenia, ale aj na vkladanie a vyberanie dát. Operácia s databázou prebieha pomocou vytvorených SQL dotazov. V metódach je najskôr pomocou prijatých parametrov vytvorený požadovaný dotaz a následne je spustení použitím JDBC.

---

<sup>3</sup>Viz <http://developers.itextpdf.com/downloads>.

<sup>4</sup>Viz <https://www.microsoft.com/en/server-cloud/products/sql-server-editions/sql-server-express.aspx>.



## Kapitola 6

# Testovanie

Táto kapitola popisuje spôsob testovania aplikácií na reálnom zariadení. V kapitole sú popísané testy palubného dielu automobilu, na ktorom bola testovaná použiteľnosť aplikácie v praxi. Testy zobrazujú grafy hodnôt namerané a vytvorené výslednou aplikáciou. Testované boli tri tlačidlá palubnej dosky. Každý test prebiehal približne tridsať minút, počas ktorých sa vykonalo približne 300 meraní. Tento čas sa ukázal ako postačujúci pre ukázanie funkčnosti aplikácie pri testoch na reálnom zariadení.

Obrázky v tejto kapitole sú vytvorené pomocou desktopovej aplikácie a nezobrazujú čas merania. Pri použití desktopovej aplikácie je čas merania zobrazený pod grafom. Tento čas slúži pre detekciu konkrétneho času vzniknutej chyby.

### 6.1 Testy nepriaznivých vplyvov na meranie

Test bol zameraný na ukázanie funkčnosti aplikácie a presnosť merania pri nepriaznivých vplyvoch spôsobených narušením chodu aplikácie. Pri reálnom nasadení aplikácie na robotickom pracovisku môže dôjsť k rôznym problémom. Jedným je napríklad kontakt človeka a robota. Tento problém môže nastať v extrémnych situáciách. Nepredpokladá sa častá konfrontácia robota s človekom alebo iným predmetom. Aj napriek tomu v takomto prípade musí robot pracovať maximálne bezpečne a svoju činnosť pozastaviť. Cieľom testu bolo ukázať schopnosť robotickej aplikácie reagovať na podnety z okolia, a zároveň ukázať faktory vplývajúce na meranie.

**Začiatok testu:** 12:50 hod.

**Koniec testu:** 13:10 hod.

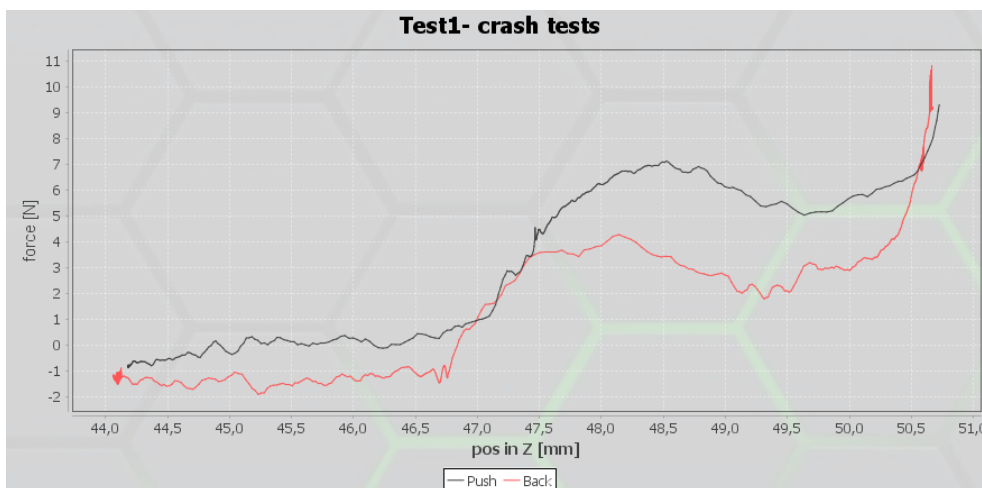
**Testované tlačidlo:** tlačidlo ručnej brzdy na palubnej doske

**Prevedené testy:**

1. Správanie pri konflikte
2. Prázdne stlačenie
3. Prekážka pri stlačení

Na obrázku 6.1 je zobrazené jedno meranie tlačidla ručnej brzdy. Graf znázorňuje závislosť sily pôsobiacej na tlačidlo pri stlačení a uvoľnení. V grafe je na ose x označená pozícia robota meraná v osi z (tzn. pozícia v súradnom systéme robota, ktorá zachytáva dráhu pri stlačení

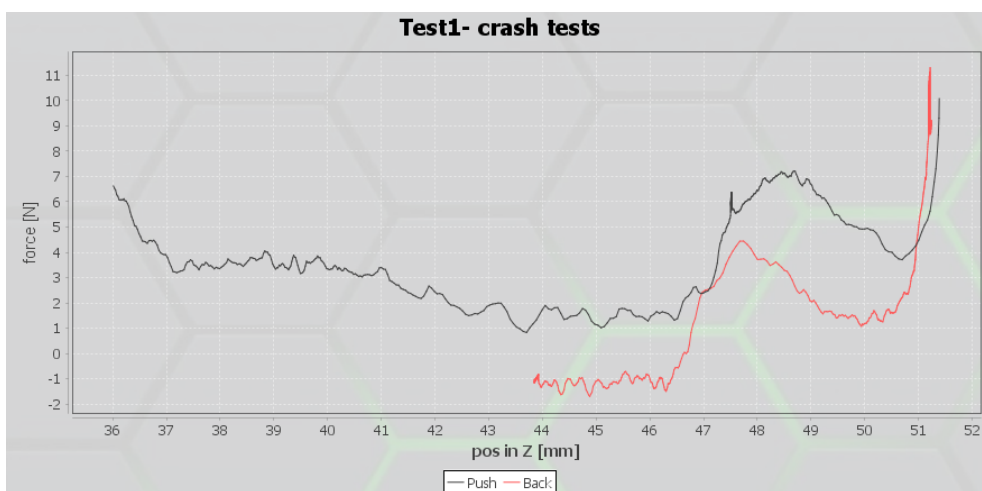
a uvoľnení tlačidla), na ose y je sila pôsobiaca na tlačidlo. Čierny graf znázorňuje sily pri stlačení, červený sily vykonané pri uvoľnení tlačidla. Podľa technických štandardov je sila potrebná na stlačenie testovaného tlačidla približne  $6 \pm 1.5$  N. Pri zobrazení grafu vo vytvorenej desktopovej aplikácii je možné vidieť silu 6.938 N, pri ktorej dôjde k prekonaniu silovej bariéry, čo spôsobí zopnutie tlačidla. Graf uvoľnenia grafu približne kopíruje graf stlačenia. Mierny posun po osi y môže byť spôsobený váhou stláčacieho predmetu, upevneného na ramene robota.



Obr. 6.1: Meranie tlačidla ručnej brzdy.

### 6.1.1 Správanie pri kontakte

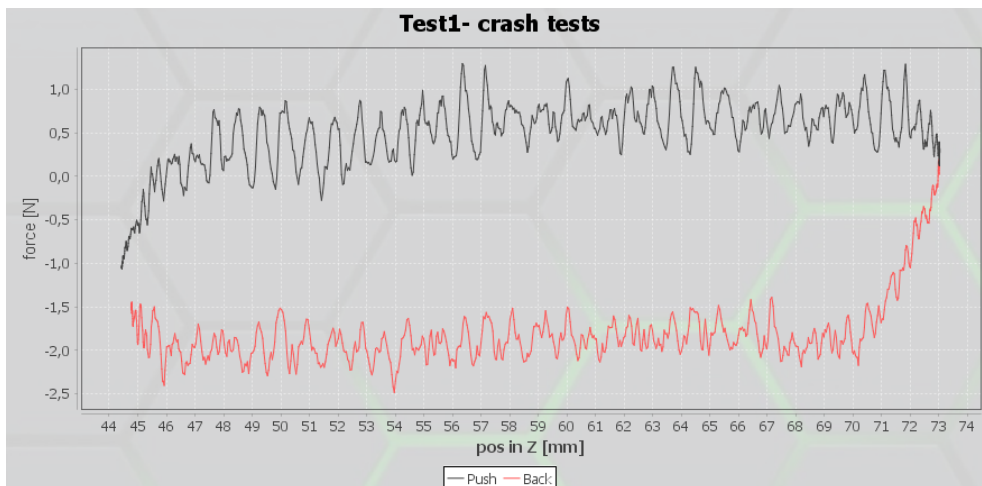
V čase 12:58 hod. bol robot úmyselne zastavený ľudským pričinením. Pri dotyku človeka s ramenom robota, robot bezpečne zastavil. V čase 12:58:10 hod. bolo namerané meranie na obrázku 6.2. Z obrázku vidno zmenu grafu v dôsledku vychýlenia z dráhy. Po opätovnom spustení sa robot vrátil do správnej pozície a meranie dokončil, o čom svedčí druhá polovica grafu.



Obr. 6.2: Meranie tlačidla ručnej brzdy pri konflikte s robotom.

### 6.1.2 Prázdne stlačenie robota

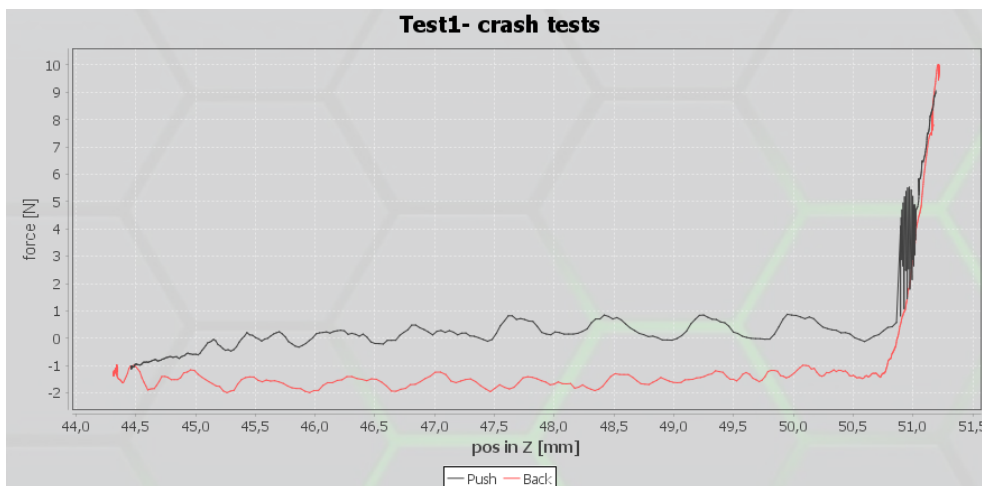
Pri druhom teste bol počas testovania odobraný testovaný diel. Výsledok merania je zobrazený na obrázku 6.3. Robot počas pohybu nezaznamenal žiadny odpor, čiže nenašiel prekážku, v dôsledku čoho sila na grafe kmitá okolo nuly. Záporné hodnoty sily pri spätnom meraní môžu byť spôsobené váhou pripevneného zariadenia na stláčanie.



Obr. 6.3: Meranie počas odobrania dielu.

### 6.1.3 Prekážka pri stlačení

Počas testu bolo meranie schválne narušené poškodením tlačidla. Obrázok 6.4 zobrazuje meranie z času 13:10:37 hod. Test simuloval zaseknutie tlačidla v maximálnej polohe. V prípade kedy je tlačidlo zaseknuté nie sú namerané žiadne sily (kmitanie okolo nuly), v momente kedy robot narazí na stlačené tlačidlo sila vystúpi do maxima (približne 10 N).



Obr. 6.4: Meranie pri stlačenom tlačidle.

## 6.2 Testy tlačidiel palubnej dosky

V druhom teste boli testované jednotlivé tlačidlá dielu. Pri každom testovanom tlačidle bola spustená robotická aplikácia približne na tridsať minút. Počas testovania boli sledované zmeny hodnôt v jednotlivých meraniach. Cieľom bolo sledovať správanie aplikácie a otestovať možnosť využitia v praxi.

### Prevedené testy:

1. Testovanie tlačidla ručnej brzdy
2. Testovanie tlačidiel auto-hold a lock/unlock

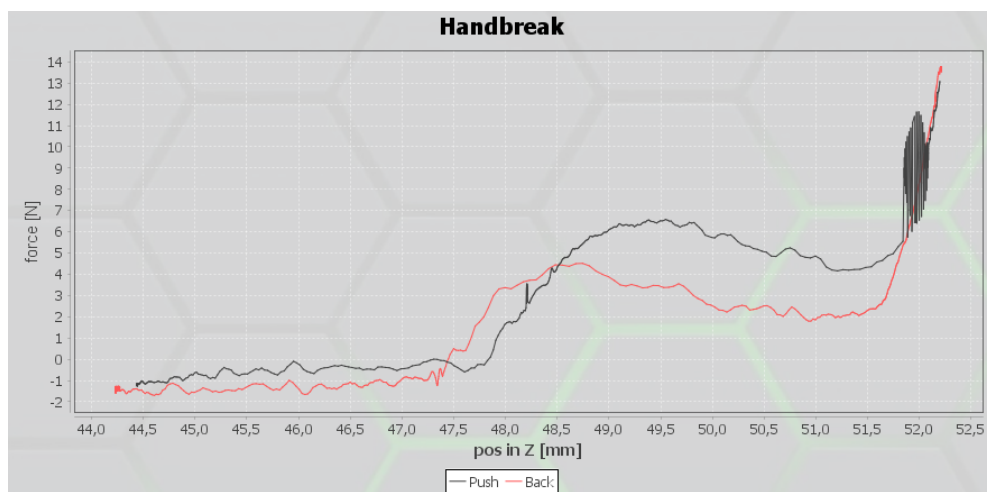
### 6.2.1 Testovanie tlačidla ručnej brzdy

**Začiatok testu:** 13:40 hod.

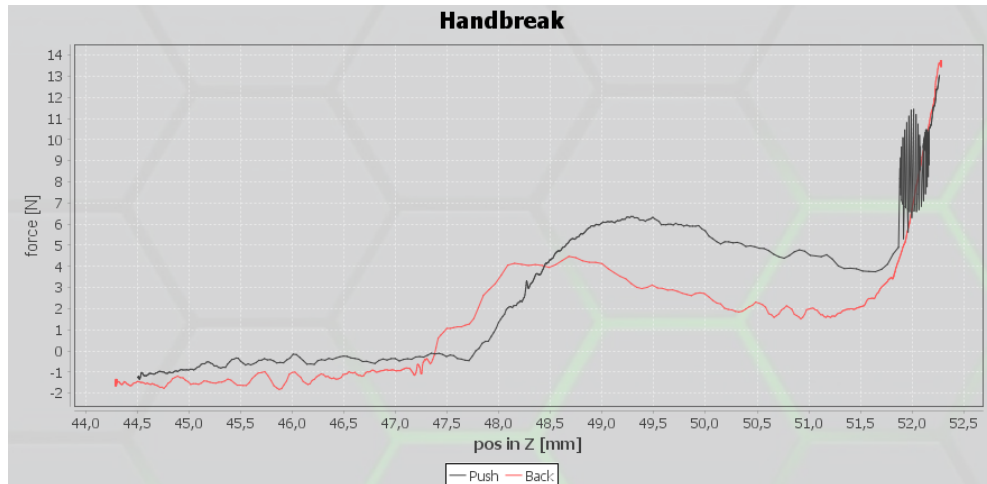
**Koniec testu:** 14:10 hod.

**Sila nutná pre stlačenie tlačidla:**  $6 \pm 1.5$  N

Počas testovania bolo vytvorených približne 250 meraní. Obrázok 6.5 zobrazuje hodnoty jedného merania zachyteného na začiatku testovania, obrázok 6.6 je vytvorený pred koncom testovania v čase 14:07:40 hod. Na grafoch vidno prekonanie sily približne v hodnote 6 N, čo zodpovedá technickej špecifikácii tlačidla. Počas celého testovania sú hodnoty grafov veľmi podobné z čoho možno predpokladať správnu funkčnosť tlačidla.



Obr. 6.5: Začiatok testovania tlačidla ručnej brzdy.



Obr. 6.6: Koniec testovania tlačidla ručnej brzdy.

## 6.2.2 Testovanie tlačidiel auto-hold a lock/unlock

**Začiatok testu:** 11:50 hod.

**Koniec testu:** 12:40 hod.

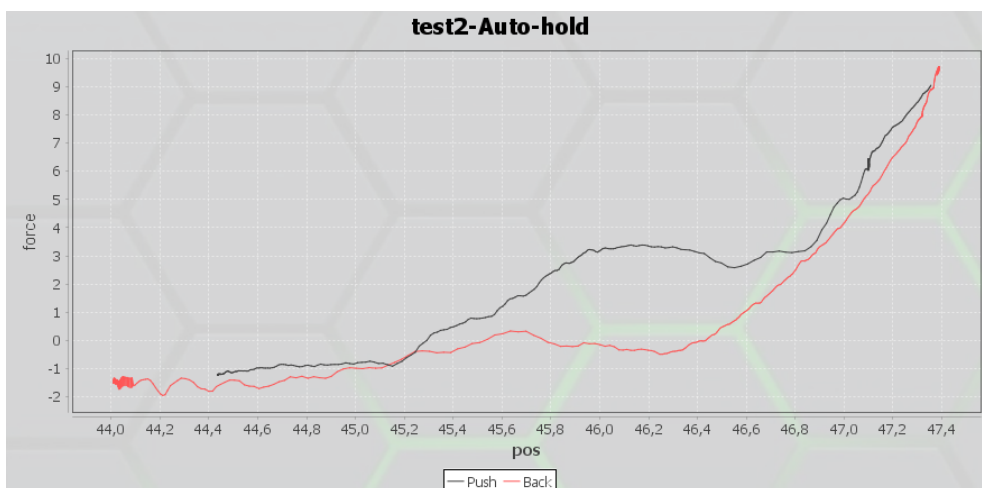
**Sila nutná pre stlačenie tlačidla:**  $3 \pm 0.7 \text{ N}$

Tlačidlá auto-hold a lock/unlock (zamknutie odomknutie dverí) majú rovnaký tvar a vlastnosti z tohoto dôvodu bolo ich testovanie zhrnuté do jedného testu. Grafy znázornené na obrázkoch 6.7 a 6.8 znázorňujú namerané hodnoty pre tlačidlá auto-hold a lock/unlock. Spustenie tlačidiel nastáva v hodne približne 3 N, čo odpovedá špecifikácii. V oboch prípadoch je mierne skreslený červený graf, ktorý znázorňuje uvoľnenie tlačidla. Táto nepresnosť môže byť spôsobená váhou nástroja na stláčanie. Testovanie tiež ukázalo citlivosť merania na upevnenie tlačidla. Pre dosiahnutie presných výsledkov musí byť zariadenie pevne upevnené na tvrdej podložke, aby nedochádzalo k pruženiu alebo šmýkaniu.

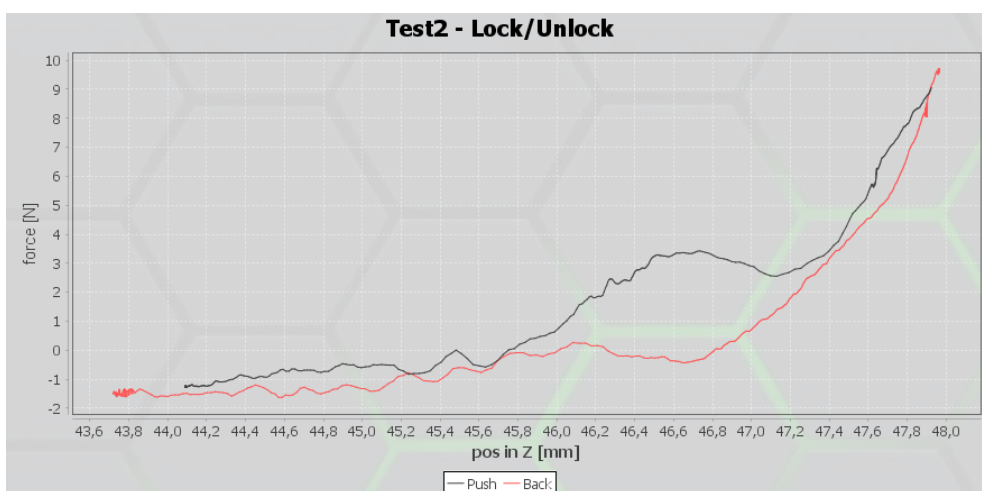
Počas testovania bola do jedného merania vnesená chyba. Chyba merania bola zaznamenaná pri testovaní tlačidla lock/unlock v čase 12:12:07 hod. Záznam, ktorý bol vytvorený pomocou aplikácie je priložený ako príloha B.

## 6.3 Záver testovania

Testovanie dokázalo funkčnosť aplikácie pri testovaní mechanických častí automobilov. Výsledný produkt dokáže zbierať a zobrazovať sily zachytené pri testovaní tlačidiel palubných dosiek. Pre korektné vyhodnotenie zobrazených grafov je potrebné poznať presnú špecifikáciu testovaných výrobku. Prevedené testy ukázali citlivosť merania na zmeny okolitého prostredia. Pre dosiahnutie presných výsledkov je nutné pevné uchopenie testovaného produktu a správne nastavenie stláčacieho nástroja upevneného na konci robotického ramena. Testovanie tiež ukázalo citlivosť merania pri zmenách testovacej pozície. Navádzací režim robota LBR iiwa nedisponuje dostatočnou jemnosťou a presnosťou. Pri učení robota vznikol problém navádzania robota do rovnakej pozície. Túto vlastnosť však nemožno ovplyvniť vytvorenou aplikáciou.



Obr. 6.7: Testovania tlačidla auto-hold.



Obr. 6.8: Testovanie tlačidla lock/unlock.

# Kapitola 7

## Záver

Táto práca popisuje vývoj aplikácie, zadanej spoločnosťou Matador Industires, ktorá slúži pre testovanie mechanických častí automobilov pomocou robota LBR iiwa. V práci bol popísaný prehľad a vývoj kolaboratívnych robotov a zároveň popis návrhu a vývoja spomínanej aplikácie. Prvá časť popisuje kolaboratívne roboty, ich prínos a možnosť nasadenia v automatizačnom priemysle. Najväčšia pozornosť bola venovaná robotu LBR iiwa, pre ktorého je vytvorená výsledná aplikácia. V druhej časti práce bol popísaný návrh a implementácia aplikácie. Vytvorený produkt pozostáva z dvoch častí, ktoré spolu spolupracujú. Robot je riadený pomocou robotickej aplikácie implementovanej v jazyku Java. Aplikácia cyklicky testuje tlačidlá palubných dosiek. Pri každom meraní sú zaznamenané sily pôsobiace na tlačidlá pri stlačení a uvoľnení. Počas celého chodu robotická aplikácia komunikuje s desktopovou aplikáciou. Desktopová aplikácia spracováva, ukladá a zobrazuje prijaté dáta. Po vykonaní veľkého počtu meraní možno rozhodnúť, pomocou desktopovej aplikácie, o správnej funkcionalite testovaného tlačidla palubnej dosky. Testy popísané v poslednej časti práce, testujú funkcionalitu aplikácií na reálnom zariadení. Výsledky testov ukazujú splnenie požiadavkov na výslednú aplikáciu.

Testovanie mechanických častí, palubných dosiek alebo ovládacích panelov bolo doteraz vykonávané prevažne mechanicky pracovníkmi. Vytvorené dielo je v súčasnej dobe, spolu s robotom, nasadené do praxe. Nasadenie aplikácie prispelo k zjednodušeniu práce v oblasti testovania.

Väčší rozvoj robotickej aplikácie môže byť prínosom v rôznych odvetviach automatizačného priemyslu. Použitie aplikácie môže odstrániť časť stereotypnej činnosti, ktorú vykonáva pracovník. Do budúca je aplikáciu možné rozšíriť na iné činnosti napríklad testovanie krútiacich mechanizmov, prevodoviek alebo iné.

# Literatúra

- [1] *ABB* [online]. [cit. 2016-3-30]. Dostupné z: <http://new.abb.com/products/robotics/yumi>.
- [2] *Dictionary.com* [online]. [cit. 2016-3-30]. Dostupné z: <http://www.dictionary.com/>.
- [3] *Rethink Robotics* [online]. [cit. 2016-3-30]. Dostupné z: <http://www.rethinkrobotics.com/build-a-bot/>.
- [4] *Selic Industriedesign* [online]. [cit. 2016-3-30]. Dostupné z: <http://selic.de/project/2B9qBfgRoFxT6dGSA>.
- [5] *IML Control Endineering : Kuka goes soft on control* [online]. júl 2010 [cit. 2016-3-30]. Dostupné z: <http://www.controlengeurope.com/article/35470/Kuka-goes-soft-on-control.aspx>.
- [6] *Roboty a robotická zařízení - Požadavky na bezpečnost průmyslových robotů : Část 2: Systémy robotů a integrace*. Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012.
- [7] *Roboty a robotická zařízení - Požadavky na bezpečnost průmyslových robotů : Část 1, Roboty*. Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012.
- [8] *Robotiq* [online]. 2014 [cit. 2016-3-30]. Dostupné z: <http://blog.robotiq.com/bid/71048/New-Collaborative-Robot-called-Roberta-Your-New-Best-Friend>.
- [9] *Fanuc* [online]. 2015 [cit. 2016-3-30]. Dostupné z: <http://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/spolupracuj%C3%ADc%C3%AD-cr35ia>.
- [10] *Universal Robots* [online]. 2015 [cit. 2016-3-30]. Dostupné z: <http://www.universal-robots.com/products/help-me-choose/>.
- [11] *Nextage* [online]. 2016 [cit. 2016-3-30]. Dostupné z: <http://nextage.kawada.jp/en/>.
- [12] Bakaguer, C. *Why Humanoid Robots* [online]. september 2006 [cit. 2016-3-30]. Dostupné z: [http://www.robot.uji.es/documents/iurs06/documents/Balaguer\\_b.pdf](http://www.robot.uji.es/documents/iurs06/documents/Balaguer_b.pdf).
- [13] Bélanger-Barrette, M. *Collaborative Robot Ebook*. 6. robotiq.com, október 2015. Dostupné z: [https://mycourses.aalto.fi/pluginfile.php/144202/mod\\_resource/content/1/Yhteisk%C3%A4ytt%C3%B6robotiikka.pdf](https://mycourses.aalto.fi/pluginfile.php/144202/mod_resource/content/1/Yhteisk%C3%A4ytt%C3%B6robotiikka.pdf).
- [14] Bélanger-Barrette, M. What does collaborative robot mean? [online]. august 2015 Dostupné z: <http://blog.robotiq.com/what-does-collaborative-robot-mean>.



- [15] Bélanger-Barrette, M. What is cobot? [online]. máj 2015 Dostupné z: <http://blog.robotiq.com/what-is-a-cobot>.
- [16] Božek, P.; Barborák, O. *Špecializované robotické systémy*. druhé. Oty Synka Poruba 1847/23, Ostrava: Nakladatelství Ámos, 2011. ISBN 978-80-904766-8-4.
- [17] Cao, U. Y.; Fukunaga, A. S.; Kahng, A. B. : Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, , č. 4, 1997: s 1–23.
- [18] History. *IFR* [online]. 2016 [cit. 2016-3-30]. Dostupné z: <http://www.ifr.org/history/>.
- [19] Kalaš, V. : Tridsať rokov svetovej robotiky. *AT&P journal*, jún 2004.
- [20] Kemp, C. C.; Fitzpatrick, P.; Hirukawa, H.; aj. *Handbook of Robotics : Humanoids*. Berlin Heidelberg: Springer-Verlag, 2008. 59, s. 1307-1329. ISBN 978-3-540-23957-4.
- [21] KUKA Robotics : *LBR iiwa Product brochure*. február 2015. Dostupné z: [http://www.kuka-robotics.com/res/sps/9cb8e311-bfd7-44b4-b0ea-b25ca883f6d3\\_LBR\\_iiwa\\_Product\\_brochure\\_EN.pdf](http://www.kuka-robotics.com/res/sps/9cb8e311-bfd7-44b4-b0ea-b25ca883f6d3_LBR_iiwa_Product_brochure_EN.pdf)
- [22] Mařík, V.; a iní : *Průmysel 4.0* [online]. február 2016 Dostupné z: <http://kzps.cz/wp-content/uploads/2016/02/kzps-cr.pdf>.
- [23] Mataric, M. J. *Interaction and Intelligent Behavior* Massachusetts Institute of Technology, máj 1994.
- [24] Rethink Robotics : *Baxter*. október 2015. Dostupné z: [http://cdn-staging.rethinkrobotics.com/wp-content/uploads/2015/11/Packaging\\_Datasheet\\_Oct2015.pdf](http://cdn-staging.rethinkrobotics.com/wp-content/uploads/2015/11/Packaging_Datasheet_Oct2015.pdf)
- [25] Singht, B.; Sellappan, N.; P., K. : Evolution of Industrial Robots and their Applications. *IJETAE*, roč. 3, č. 5, máj 2013, ISSN 2250-2459.
- [26] techcentury.com. *New Fanuc robot disigned to be collaborative* [online]. jún 2015 [cit. 2016-3-30]. Dostupné z: <http://techcentury.com/2015/06/30/new-fanuc-robot-designed-to-be-collaborative>.

# Prílohy

## Zoznam príloh

<b>A</b>	<b>Obsah DVD</b>	<b>40</b>
<b>B</b>	<b>Záznam o chybnom meraní vytvorený v desktopovej aplikácii</b>	<b>41</b>
<b>C</b>	<b>Užívateľský Manuál</b>	<b>42</b>
	C.1 Požiadavky aplikácie: . . . . .	42
	C.2 Preklad a spustenie . . . . .	42
	C.3 Ovládanie . . . . .	42

# Príloha A

## Obsah DVD

- `bp_xstude22.pdf` – text písomnej práce vo formáte PDF
- `tex/` – zdrojové súbory tejto písomnej práce v jazyku  $\text{\LaTeX}$
- `manual.pdf` – manuál pre preklad, spustenie a prácu s vytvorenými aplikáciami
- `desktop_part/` – zdrojové súbory desktopovej časti aplikácie
- `robotic_part/` – zdrojové súbory robotической časti aplikácie
- `video/` – sprievodné video ukazujúce prácu s aplikáciou

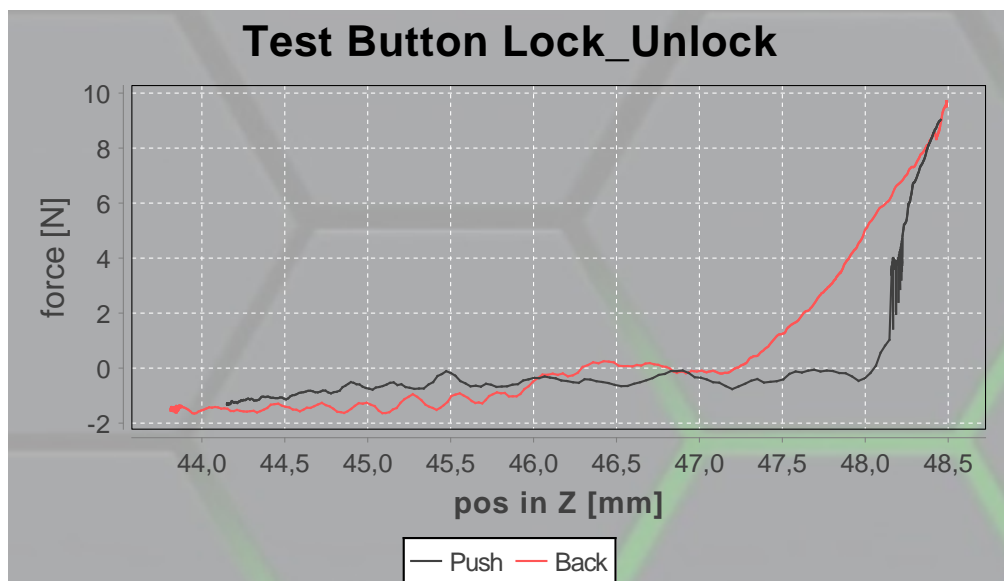
## Príloha B

# Záznam o chybnom meraní vytvorený v desktopovej aplikácii

Portion 1: Part1-daspboard

Description: First testing part. Contains 3 buttons: Auto-Hold, Handbreak, Lock-Unlock.

Date/Time: 2016-04-29 12:12:07



# Príloha C

## Užívateľský Manuál

### C.1 Požiadavky aplikácie:

- Robot LBR iiwa od spoločnosti KUKA
- OS Windows 7 a vyšší
- MS SQL Server databáza
- JRE 1.8

### C.2 Preklad a spustenie

#### Robotická časť

Robotickú aplikáciu možno nahráť do robota LBR iiwa prostredníctvom programu SunriseWorkbench od spoločnosti KUKA. V prostrední stačí prekopírovať zdrojové súbory z adresára `robotic_part/` do nového alebo existujúceho projektu. Následne je nutné nastaviť cestu pre preklad knižníc z adresára `robotic_part/lib/` a vytvoriť `Frame-i Quality/Home` a `Quality/PushUP`.

#### Desktopová časť

Desktopovú aplikáciu možno preložiť a spustiť z príkazovej riadky v priečinku `desktop_part/`:

- `ant compile`: kompiluje zdrojové súbory
- `ant run`: spustí aplikáciu
- `ant clean`: vymaže vytvorené súbory

### C.3 Ovládanie

Pre správny beh programu je požadované priame spojenie robota s počítačom, na ktorom beží desktopová aplikácia. Pred spustením desktopovej aplikácie je nutné nastaviť informácie o databáze v súbore `config.properties`. Po spustení sa aplikácia pokúsi pripojiť k databáze a vytvoriť server pre prijímanie dát od robota. Ak prebehla inicializácia správne aplikácie vypíše správu „Ready to collect data“.

Pri prvom spustení je potrebné zadať meno súčiastky a vytvoriť záložku z grafom ktorá bude zobrazovať meranie.

- **File -> Add button:** umožňuje pridať nové testované tlačidlo
- **File -> Remove button:** odstráni záložku testovaného tlačidla
- **Edit -> Add new part:** umožňuje pridať novú súčiastku
- **Settings:** obnovenia serveru alebo pripojenia k databáze

Ak sa po pridaní novej záložky vytvorený prázdny graf v databáze nie sú uložené žiadne merania o testovanom tlačidle.

Tlačidlá šípok v aplikácii slúžia pre zobrazenie historicky nameraných hodnôt. Pri klikaní na šípky je možné zobrazovať ďalšie merania. V prípade kedy sa už v databáze nenachádzajú ďalšie merania aplikácia sa prepne do automatického režimu. V tomto režime sú zobrazované aktuálne prijaté dáta od robota. Po kliknutí na tlačidlo Export je možné exportovať záznam, o aktuálne zobrazenom meraní, do formátu PDF.

Po spustení robotickéj aplikácie je nutné zvoliť číslo tlačidla, ktoré je aktuálne testované. Robot sa presunie do počiatočnej pozície a zobrazí dialógové okno na KUKA SmartPAD-e. Operátor môže vybrať možnosť navádzania robota alebo zapnúť testovanie tlačidla, ktoré bolo testované pri poslednom spustení aplikácie. V navádzacom režime je nutné držať šedé tlačidlo na ramene robota. Naučenie nového bodu je možné stlačením zeleného tlačidla. Po naučení nového bodu robot vykoná jedno testovacie meranie a zobrazí výsledok na obrazovku KUKA SmartPad-u. Následne môže operátor opakovať učenie nového bodu alebo spustiť testovanie tlačidla. Počas testovania robot odosiela dáta desktopovej aplikácii, ktorá dáta ukladá a zobrazuje. Robotickú aplikáciu je možné ukončiť prostredníctvom KUKA SmartPAD-u.