

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2021

Jan Fridrich

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**MOŽNOSTI PROSTŘEDÍ MATLAB PRO OPTIMALIZACI
A PROGRAMOVÁNÍ ROBOTŮ STÄUBLI**

Jan Fridrich

Bakalářská práce

2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jan Fridrich**
Osobní číslo: **I17063**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Téma práce: **Možnosti prostředí MATLAB pro optimalizaci a programování robotů Stäubli**
Zadávající katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cíl: Seznámit se s programem Optimize Lab a vytvořit ukázkovou aplikaci využívající možnosti SW. Zjistit a otestovat možnost komunikace případně programování robotů Stäubli z vývojového prostředí MATLAB.

Obsah teoretické části: Robotická ramena, kategorie pro různé aplikační oblasti, způsoby programování, optimalizace programu robota.

Obsah implementační části: Popis robotů Stäubli, programování, popis Optimize Lab, návrh ukázkové aplikace použití Optimize Lab, testování možnosti komunikace a programování robotů Stäubli z prostředí MATLAB, zhodnocení výsledků.

Rozsah pracovní zprávy: **50**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

HAVELKA, Martin a Veronika STOFFOVÁ. Robotika – stavba a programování robotů. Olomouc: Univerzita Palackého v Olomouci, 2017. ISBN 978-80-244-5194-7.
ZÁDA, Václav. Robotika: matematické aspekty analýzy a řízení. Liberec: Technická univerzita v Liberci, 2012. ISBN 978-80-7372-882-3.
SCARA, 6-axis, FAST picker and collaborative robots / Stäubli [online]. Stäubli [cit. 6. 11. 2019]. Dostupné na: <<https://www.staubli.com/en/robotics/>>. Simulator for Industrial robots and offline programming – RoboDK [online]. RoboDK [cit. 6. 11. 2019]. Dostupné na: <<https://robodk.com/>>.

Vedoucí bakalářské práce: **Ing. Daniel Honc, Ph.D.**
Katedra řízení procesů

Datum zadání bakalářské práce: **17. prosince 2019**
Termín odevzdání bakalářské práce: **7. května 2020**

L.S.

Ing. Zdeněk Němec, Ph.D.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 20. ledna 2020

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 1. 8. 2021

Jan Fridrich

Poděkování

Rád bych tímto poděkoval panu Tomáši Burianovi ze Stäubli za ochotu a spolupráci.

V Pardubicích dne 1. 8. 2021

Jan Fridrich

ANOTACE

Cílem práce je najít možnosti řízení robota Stäubli TX2 z prostředí Matlab a vytvořit ukázkovou aplikaci pro řízení. Dále pak seznámit se s nástrojem Optimize Lab pro optimalizaci robotů vyvinutou společností Stäubli. Teoretická část je zaměřená na seznámení se s roboty. V praktické části je popsán vzhled a funkce aplikace a samotné programové řešení. Následně jsou představeny možnosti Optimize Lab na ukázkové aplikaci.

KLÍČOVÁ SLOVA

Matlab, Stäubli, SRS, TX2, socket

TITLE

*OPTIMIZATION AND PROGRAMMING OF STÄUBLI ROBOTS IN MATLAB
ENVIROMENT*

ANNOTATION

The first objective of this work is find Matlab options for controlling/programming Stäubli robot TX2 and create sample application. Next will be introduced the Optimize Lab tool for optimizing the robots developed by Stäubli. The theoretical part is focused on getting acquainted with the robots. The practical part describes the appearance of the application and its function and the software solution itself. Subsequently, the possibilities of Optimize Lab for sample application are presented.

KEYWORDS

Matlab, Stäubli, SRS, TX2, socket

OBSAH

	Seznam zkratk	9
	Seznam ilustrací	10
	Seznam tabulek	11
	Úvod	12
1	Průmyslová robotika	13
1.1	Druhy vyráběných robotů	13
1.1.1	Podle konstrukce	13
1.1.2	Podle využití	16
2	Stäubli	17
2.1	Druhy robotů a využití	17
2.1.1	Popis šestiosého robota TX2	18
3	Programování robotů STÄUBLI	20
3.1	SRC	20
3.1.1	Teach pendant a jeho obsluha	20
3.2	SRS studio	21
3.2.1	Obsluha	21
4	Matlab	24
4.1	Grafické rozhraní	24
4.1.1	GUIDE	24
4.1.2	GUI pomocí m-File	24
4.1.3	App Designer	24
4.2	Optimize Lab	25
5	Komunikační rozhraní	26
5.1	Typy komunikací	26
5.1.1	Socket	26
6	Návrh aplikace	28
6.1	Grafické rozhraní	28
6.2	Popis funkcí aplikace	29
7	Programové řešení	30
7.1	Matlab	30
7.1.1	Socket	30
7.1.2	Další funkce	30

7.2	SRS studio	31
7.2.1	Socket	32
7.2.2	Pohybové funkce	33
7.2.3	Ostatní funkce	34
7.2.4	Určení bodů	35
8	Optimize Lab	36
8.1	Záznam a konfigurace	36
8.2	Délka cyklu a životnost	37
8.3	Ukázka parametrů	38
8.4	Simulace	39
8.5	Ukázka špatné optimalizace a možné řešení	40
8.6	Využití addonů	41
9	Závěr	42
	Použitá literatura	43
	Přílohy	46

SEZNAM ZKRATEK

AVG	Automated guided vehicles
CAD	Computer aided design
ESD	Eletrostatic discharge
FTP	File transfer protocol
GUI	Graphical User Interface
GUIDE	Graphical User Interface Development Environment
HE	Hygienic and humid enviroments
I/O	Input/Output
PLC	Programmable Logic Controller
SCARA	Selective Compliance Articulated Robot Arm
SCR	Super cleanroom
SRC	Stäubli robotics controls
SRS	Stäubli robotics suite
TCP/IP	Tramsmission control protocol/Intertnet protocol
UDP	User datagram protocol

SEZNAM ILUSTRACÍ

Obrázek 1.1 – SKARA (Epson, 2021)	13
Obrázek 1.2 – Delta robot (AutoCont Control Systems, 2021)	14
Obrázek 1.3 – ABB YuMi (ABB, 2021)	15
Obrázek 1.4 – Kloubový robot (Berger Positec, 2021)	15
Obrázek 1.5 – Svařovací robot (Ferier, 2021)	16
Obrázek 2.1 – SCARA Stäubli (Stäubli, 2021)	17
Obrázek 2.2 – TX2 (Stäubli, 2021)	18
Obrázek 3.1 – Teach pendant v emulátoru	20
Obrázek 3.2 – SRS studio	22
Obrázek 3.3 – Karty Data a Geometry	22
Obrázek 3.4 – Grafické rozhraní	23
Obrázek 6.1 – Tvorba kanálu, vytvoření serveru	28
Obrázek 6.2 – Ovládací aplikace	29
Obrázek 7.1 – Socketový server v SRS studiu	32
Obrázek 7.2 – Povolení užívání ventilů	34
Obrázek 7.3 – Uživatelské rozhraní	34
Obrázek 7.4 – Nastavení souřadnic bodů	35
Obrázek 8.1 – Úvodní okno, nastavení konfiguračního souboru	36
Obrázek 8.2 – Spuštění záznamu, zobrazení v transfer manageru optimalizace	37
Obrázek 8.3 – Nastavení doby cyklu	37
Obrázek 8.4 – Ukázka životnosti	38
Obrázek 8.5 – Zobrazitelné parametry	38
Obrázek 8.6 – Točivý moment převodovky	39
Obrázek 8.7 – Export do SRS studia, umístění záznamu	39
Obrázek 8.8 – Simulace pohybu	40
Obrázek 8.9 – Vzor zatížení v Optimize Lab	40
Obrázek 8.10 – Vzor zatížení v simulaci	41

SEZNAM TABULEK

Tabulka 2.1.1 – Charakteristika jednotlivých variant robota TX2 (TX2 katalog, 2021) ... 19

ÚVOD

Použití robotů v dnešní době je nedílnou součástí našeho života. Nacházejí se v našem okolí a často nám usnadňují práci, aniž bychom si to uvědomovali. Nahrazují člověka v pracích, které jsou fyzicky náročné, nebo při kterých je kladen důraz na preciznost práce. Člověk si už od nepaměti veškerou práci rád usnadní, a proto se roboty neustále rozšiřují do dalších odvětví, a to má za následek propojování různých systémů. To však klade větší důraz i na lidi, kteří zajišťují správu robotů a kteří musí zajistit bezpečnost celého procesu, ale zároveň musí zachovat obsluhu co nejvíc intuitivní.

Cílem práce je představení možnosti řízení robota Stäubli z jiného prostředí, než je běžné, a ukázání možnosti kooperace různých prostředí pro maximální využití jejich potenciálu. Cílem je tedy navrhnout aplikaci v prostředí Matlab, která bude sloužit pro řízení robota Stäubli z řady TX2 v reálném čase. Součástí bude i vytvořený program pro samotného robota, který bude obsluhovat příchozí požadavky. Dále pak otestovat nástroj Optimize Lab pro optimalizaci pohybu robota pro zajištění co nejdelší možné provozuschopnosti.

1 PRŮMYSLOVÁ ROBOTIKA

Průmyslový robot je označení pro stroj, který je univerzální a může tak sloužit k více účelům. Dále je důležité, aby byl schopný automatické činnosti. To znamená, že zvládne samostatně fungovat bez zásahu člověka. Zároveň by měl jít program velice snadno změnit. V neposlední řadě je nutná jeho manipulační schopnost, tedy aby dokázal uchopit předmět a provádět například plnohodnotně montážní úkony (Žáček, 2018).

1.1 DRUHY VYRÁBĚNÝCH ROBOTŮ

Průmyslové roboty urychlují a zjednodušují práci v mnoha odvětvích. Podle způsobu využití je dělíme podle konstrukce a využití do několika skupin.

1.1.1 Podle konstrukce

SCARA (Selective Compliance Articulated Robot Arm) je typ průmyslového robota. Rameno je mírně poddajné ve směru X a Y, ale tuhé ve směru Z. To je výhodné pro mnoho typu montáží.

Jsou rychlejší než ostatní robotické systémy. Mají jeden podstavec, který má malý půdorys a poskytuje snadnou formu montáže. Výhodou je malé zabránění místa a nízká hmotnost. Také jsou oproti konkurenčním cenově výhodnější (Havlíček, 2018).



Obrázek 1.1 – SKARA (Epson, 2021)

Delta roboty vynikají v přesnosti a rychlosti. Využívají se v řadě odvětví, například v automobilním, chemickém, elektrotechnickém či farmaceutickém průmyslu. Výhody

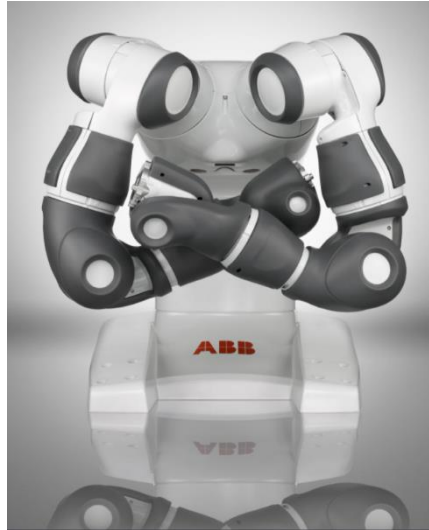
těchto robotů jsou velká univerzálnost, práce na větším pracovním prostoru nebo také poměrně velká rychlost. Dále se dělí podle potřebného zatížení, rychlosti nebo podle počtu os. Ty mohou být tříosé, čtyřosé nebo šestiosé. Robot je umístěn nad pracovní plochu, umí manipulovat s menšími lehkými předměty vysokou rychlostí (Zbiralová, 2019).



Obrázek 1.2 – Delta robot (AutoCont Control Systems, 2021)

Kartézské roboty jsou mechatronická zařízení, která se pohybují lineárně po osách X, Y a Z. Mají kovovou konstrukci, která slouží jako rám. Ten stanovuje trasy jednotlivých os, také funguje jako podpora pro zdvih při zatížení. Konstrukce může být namontována horizontálně i vertikálně, v některých případech se lze setkat i se závěsným uchycením (E-konstruktér, 2014).

Šestiosé roboty mají šest stupňů volnosti. Mohou se pohybovat dopředu a zpátky, nahoru a dolů a také umí vybočovat, naklánět se nebo otáčet se svým nástrojem. Jsou vhodné pro případ náročnějších pohybů, umožňují větší flexibilitu. Mají simulovat pohyb lidské paže. To může být, například v případě shýbání se pro součástku, kterou je potřeba zdvihnout, a umístit ji na dané místo. Jsou využívány pro paletizaci, svařování nebo také pro obsluhu strojů. Jsou poměrně náročné na programování (E-konstruktér, 2014).



Obrázek 1.3 – ABB YuMi (ABB, 2021)

Dvouramenné roboty jsou schopni pracovat velmi rychle a efektivně. Jsou vhodné pro menší montážní operace zejména v oblasti elektrotechnického průmyslu.

Kloubové roboty mohou mít systém se dvěma až deseti klouby. Mají otočný trup, biceps, rameno, předloktí a zápěstí. Tyto roboty jsou jedny z nejpoužívanějších na trhu. Jsou rychlé, flexibilní, přesné a s poměrně velkým dosahem. Mají řadu uplatnění, od jemné manipulace se součástkami po větší a těžší operace. Používají se například pro přemísťování nebo paletizace (Žáček, 2018).



Obrázek 1.4 – Kloubový robot (Berger Positec, 2021)

1.1.2 Podle využití

Svařovací roboty vynikají v precizní kvalitě svarů. Zvládají sváření elektrickým obloukem, pájení, laserové či vysokorychlostní svařování nebo řezání. Potřebují menší pracovní plochu, jsou bezpečné a kompatibilní s mnoha výrobci. Díky tomu zvyšují kvalitu výroby (Žáček, 2018).



Obrázek 1.5 – Svařovací robot (Ferier, 2021)

Lakovací roboty se nejvíce využívají v automobilovém průmyslu nebo ve stavebnictví. Používají se k lakování aut nebo nábytku. Jejich výhodou je úspora barev a vysoká kvalita nátěru (Žáček, 2018).

Paletizační roboty jsou uplatněny nejvíce ve skládání palet. Tento robot dokáže uzvednout až dvě tuny. Dále se používá k manipulaci se sádkou, sklem, obložení nebo v potravinářském odvětví (Žáček, 2018).

Kolaborativní roboty jsou užívány díky jejich vysoké preciznosti. Používány jsou při svařování, odměřování, šroubování nebo umístění předmětů. Tyto roboty mohou pracovat vedle člověka bez rizika nebezpečí (Žáček, 2018).

2 STÄUBLI

Stäubli je Švýcarská firma zabývající se mechatronickým řešením problémů ve třech odvětvích. Zabývají se fluidními a elektronickými konektory, od rychlospojek pro různé druhy energií, přes elektronické konektory pro napájení a přenos dat, až po multispojky a robotické výměny nástrojů. Dále se zabývají textilním průmyslem, kde nabízí inovativní systémy pro řešení produkce textilií. Třetím odvětvím jsou robotická řešení v průmyslu 4.0 včetně robotů, kontrolérů a softwaru (Stäubli, 2021).

2.1 DRUHY ROBOTŮ A VYUŽITÍ

Stäubli vyvíjí několik typů robotů. Prvním je robot pojmenován názvem SCARA (Selective compliance assembly robot arms). Jedná se o čtyřosého robota, na kterého jsou kladeny přísné požadavky pro práci v nejrůznějších prostředích, například jako jsou potravinářský, farmaceutický či automobilový průmysl. Je vhodný pro přesně se opakující úkony, jako je nakládání/vykládání předmětů, montáže, manipulace a třídění, a to vše ve vysokých rychlostech (Stäubli, 2021).



Obrázek 2.1 – SCARA Stäubli (Stäubli, 2021)

Dalším typem je šestiosý robot s označením TX2. Jedná se o robotické rameno, které maximálně využívá svůj pracovní prostor. Výhodou je přizpůsobení montážního umístění (strop, stěna, podlaha), které usnadňuje začlenění robota do výrobního procesu. Stejně jako SCARA, i TX2 jsou vhodná pro použití v mnoha průmyslových oblastech. Z řady TX2 je vyvíjen kolaborativní robot pod jménem TX2touch, který umožňuje bezpečnou práci s člověkem (Stäubli, 2021).



Obrázek 2.2 – TX2 (Stäubli, 2021)

Jsou také vyvíjeny mobilní roboty se jménem HelMo, které zvládají pracovat plně autonomně, nebo ve spolupráci s lidmi, se schopností navigace v pracovním prostředí. Zvládají shromažďovat materiály a následně provést práci (Stäubli, 2021).

Dalším vyvíjeným systémem jsou AVG (automated guided vehicles) roboty. Jedná se o automatizovaně naváděná vozidla pro převoz nejtěžších nákladů, která přesto vynikají svou obratností. Jsou všesměrově ovladatelná a díky inteligentnímu softwaru zajišťují vyšší bezpečnost (Stäubli, 2021).

2.1.1 Popis šestiosého robota TX2

Robot TX2 je v podobě šestiosého ramena, viz Obrázek 2.2. Rameno je typické svojí kompaktní konstrukcí, která umožňuje práci i ve sterilním prostředí. Na vnějším prostoru se nenacházejí žádné kabely ani hadice, na kterých by se mohla zachycovat kapalina a jiné částice, jelikož je všechno vedeno vnitřkem robota, který je zcela utěsněný.

TX2 se provádí v různých variantách podle potřeby použití. Dostupné jsou v HE (hygienic and humid environments), ESD (electronics compliant), SCR (super Cleanroom) a Stericlean provedení. Je tedy možné je využít jak při práci s elektronickými předměty, tak i v prostředích, která se musí udržovat sterilní. Díky kompletnímu utěsnění jim nevádí tedy ani vodní prostředí.

Tento robot se vyrábí v různých velikostech s lišícími se parametry. Vyzdvihnout lze opakovatelnost tohoto robota, která je v rozmezí $\pm 0,02 - 0,05$ mm, díky které se může řadit do světové špičky. Opakovatelnost udává, s jakou přesností se robot dokáže vrátit do stejného místa při repetitivních úkonech (Plastic Portal, 2021).

Tabulka 2.1.1 – Charakteristika jednotlivých variant robota TX2 (TX2 katalog, 2021)

Model	TX2-40	TX2-60	TX2-60L	TX2-90	TX2-90L	TX2-90XL	TX2-140	TX2-160	TX2-160L
Nosnost kg	2	4,5	3,7	14	12	7	40	40	25
Dosah mm	515	670	920	1000	1200	1450	1510	1710	2010
Opakovatelnost mm	± 0,02		± 0,03		± 0,035	± 0,04	± 0,05		
Maximální kartézská rychlost m/s	8,	8,4	11,1	10,9	11,1	11,6	9,5	10,3	12,2
Váha kg	29	52	53	114	11	119	250	260	265

Nutné je ještě zmínit, že veškeré šestiosé roboty mají implementován stejný ovladač CS9, díky čemuž je snazší implementace složitějších řešení s větším počtem robotů. Ovladač přináší pokročilé bezpečnostní funkce, které zajišťují dokonalou ochranu obsluhy a provozního zařízení. Taktéž umožňuje spolupráci člověka a robota (Stäubli, 2021).

3 PROGRAMOVÁNÍ ROBOTŮ STÄUBLI

3.1 SRC

Stäubli robotics controls (SRC) podporuje různá softwarová řešení, která umožňují programování všech robotických funkcí za použití rozdílných prostředí. Příkladem může být jazyk VAL 3, který umožňuje přístup ke všem funkcím z jediného uživatelského rozhraní, jako je teach pendant. To je výhodné pro lidi, kteří integrují robota do procesu. Dalším příkladem může být uniVAL PLC. To umožňuje programování robotů pomocí průmyslových PLC zařízení připojených přes standardní sběrnici (Stäubli, 2021).

3.1.1 Teach pendant a jeho obsluha

Teach pendant je fyzické uživatelské rozhraní umožňující přístup ke všem funkcím robota a s možností úpravy programu.



Obrázek 3.1 – Teach pendant v emulátoru

Po levé straně se nacházejí tlačítka pro hlavní menu, krok zpět, kontextová stránka a tlačítka pro ovládání digitální výstupů. Na horní straně se nachází nouzové tlačítko pro zastavení (není k náhledu na Obrázek 3.1), tlačítko pro napájení ramene, move/hold tlačítko, tlačítka pro změnu rychlost, tlačítka pro manuální pohyb a na zadní straně bezpečnostní třípolohové tlačítko.

Na displeji se nachází základních 6 dlaždic. Dlaždice pro pohyb umožňuje učení pozic bodů a manuální pohyb. VAL3 umožňuje otevírat, spouštět a ukládat aplikace. Logger obsahuje záznam událostí včetně všech chyb. Položka IO označuje vstupní a výstupní periférie, obsahuje všechna připojená rozhraní. Položka ROBOT zajišťuje nahrávání záznamů pohybu, provádí kalibraci a obnovu, lze tu nalézt hardwarové a softwarové informace o verzích, a poslední položka je nastavení, ve které lze najít nastavení sítě, uživatelské profily, datum a čas a nastavení jazyka.

3.2 SRS STUDIO

Stäubli robotics suite (SRS) studio je počítačový software pro vývoj programů. Software je omezen bez licence. Umožňuje náhled vytvořených programů, ale bez možnosti editace. Obsahuje online nápovědu a podporuje základní verzi 3D zobrazení trajektorií a samotného robota pomocí emulátoru. Umožňuje vytvářet uživatelské profily s různými právy přístupnými pomocí uživatelského jména a hesla.

Licencované verze se dělí na rozvoj a údržbu. Verze pro rozvoj umožňuje přístup ke všem funkcím potřebných pro realistické simulace jednoho či více robotů. To zahrnuje funkce pro automatické doplnění příkazů, kontrolu syntaxe či online odladění programu. Zároveň přidává možnosti pro importování CAD modelů.

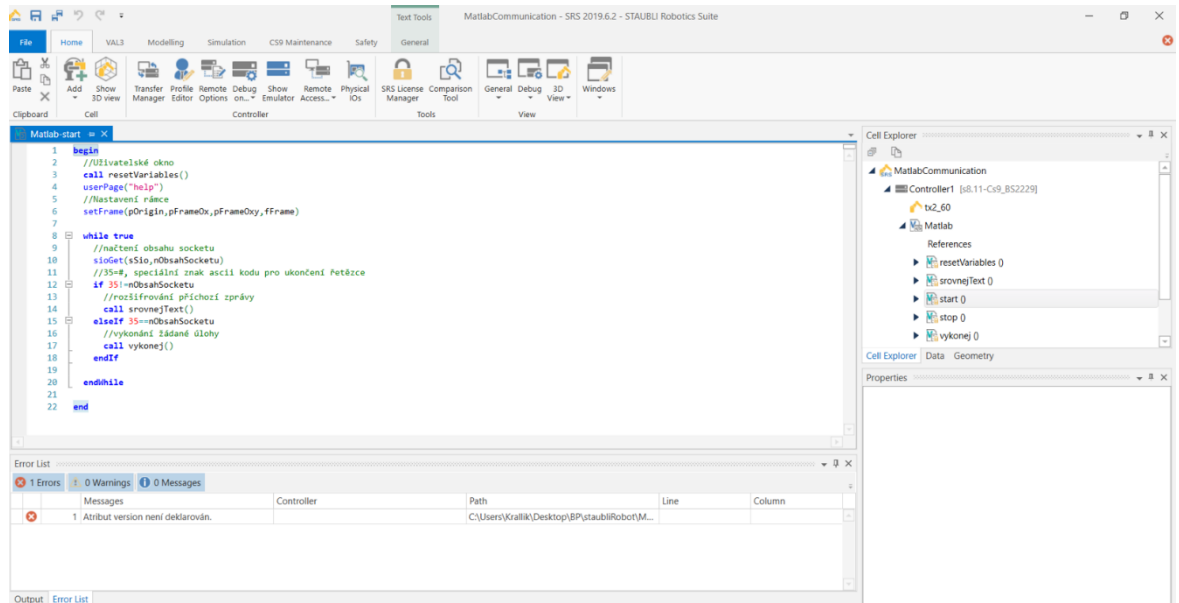
Licence pro údržbu umožňuje přístup ke všem funkcím požadovaným pro vzdálené připojení k reálnému robotu a provádění diagnostických operací včetně monitorování cyklů. Přidává přístup přes Ethernet k robotickým systémům, který zajišťuje interakci s řídicími jednotkami robota a teach pendanty (Stäubli, 2021).

3.2.1 Obsluha

Před samotným založením programu je nutné zkontrolovat aktuálně nainstalovanou verzi ovladače CS9 a případně stáhnout a nainstalovat aktuální. Následně lze spustit existující program, případně vytvořit nový. Při vytváření nového je nutné zapsat jméno a následně je na výběr ovladač. Lze stáhnout konkrétní verzi ovladače z robota, případně zvolit nový a vybrat robotické rameno, pro které se bude tvořit aplikace.

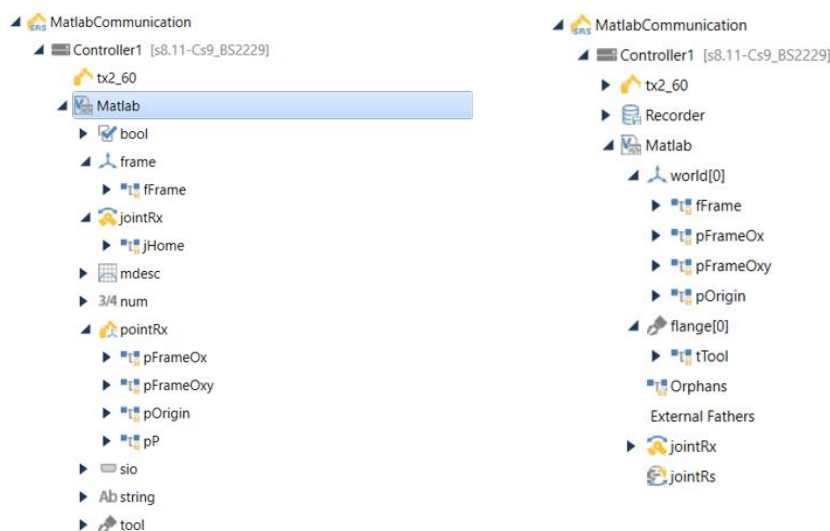
Po otevření vytvořeného programu se nám spustí základní okno. Na horní straně se nacházejí funkční pole. V kartě home lze nalézt 3D zobrazení, spuštění emulátoru, transfer manager, který zajišťuje funkci FTP serveru, díky kterému lze nahrávat existující programy

do reálného robota. Dále lze nalézt dostupná fyzická rozhraní a zjistit, zda je přítomna nějaká licence pro SRS studio.



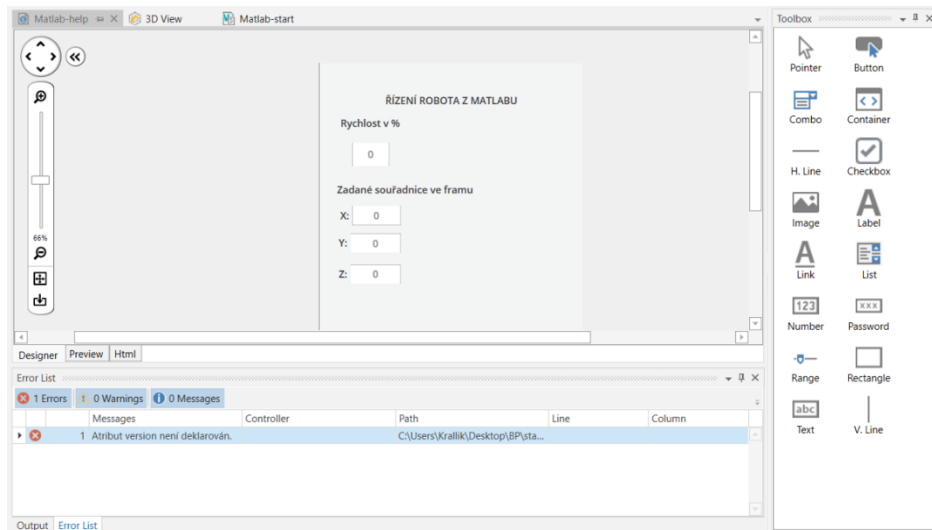
Obrázek 3.2 – SRS studio

Na kartě VAL3 lze otevírat a zakládat nové aplikace pro stejný ovladač. Také se zde spouští simulace a umožňuje se kontrola správné syntaxe. Karta modelling se věnuje importu 3D prvků do simulace. Na spodní liště se nachází veškeré informace k vyskytnutým chybám při nahrávání či spuštění programu. Po pravé straně se nachází adresář se třemi lištami. Cell Explorer obsahuje všechny podprogramy a uživatelská grafická rozhraní. V kartě data se nacházejí veškeré proměnné, a zároveň se tam i vytvářejí. Ve třetí kartě Geometry se nacházejí nahrávky záznamů pohybu a zároveň vytvořené nástroje a souřadnicové systémy.



Obrázek 3.3 – Karty Data a Geometry

Je možná tvorba i grafických rozhraní pro teach pendant, která mohou sloužit například jako kontrolní obrazovka pro sledování údajů. Jedná se o jednoduché intuitivní prostředí s vlastním nástrojovým panelem obsahujícím všechny dostupné prvky. Táhnutím myši lze pokládat a upravovat zvolené předměty a dvojklikem následně svázat se sledovanou proměnnou.



Obrázek 3.4 – Grafické rozhraní

4 MATLAB

Matlab (Matrix Laboratory) je programovací prostředí. Umožňuje například počítačové simulace, analýzu, vizualizace, vytváření aplikací či uživatelského prostředí. Je využíván pro vědecké a výzkumné účely napříč různými odvětvími, nejčastěji pro obory technické nebo ekonomické. Dále nabízí i knihovny funkcí, toolboxy nebo simulační prostředí Simulink. Knihovny funkcí, také nazývané aplikační knihovny, rozšiřují využití programů.

Simulink je prostředí, které využívá Matlab pro simulaci a modelování dynamických systémů. Tyto modely jsou vytvářeny ve formě blokových schémat (HUMUSOFT, 2021).

4.1 GRAFICKÉ ROZHRAŇÍ

GUI (Graphical User Interface) se skládá z grafických objektů, díky kterým je snadnější práce s daným programem. V této paletě komponent je možné nalézt například textová pole, posuvné seznamy, různá tlačítka, osy aj. Rozhraní je velmi intuitivní a díky tomu je snadné obsluhovat aplikace i bez předchozích znalostí. GUI lze vytvořit více způsoby.

4.1.1 GUIDE

GUIDE (Graphical User Interface Development Environment) je prostředí, které umožňuje rychlou a jednoduchou práci s nástroji pro tvorbu GUI. Volením jednotlivých komponentů v editoru a následným přesunutím na dané místo v návrhové ploše je možné vytvořit výsledek totožný s návrhem.

4.1.2 GUI pomocí m-File

V tomto případě je možné vytvořit GUI bez pomoci GUIDE. Soubor je tvořen díky jednomu souboru M-File, který obsahuje jak kód GUI, tak i kompletní popis grafické části (Blaška, 2008).

4.1.3 App Designer

V roce 2016 byla implementována funkce v podobě App Designeru, který umožňuje vytvářet profesionální aplikace. Vše probíhá přetažením jednotlivých komponent pro tvorbu

grafického rozhraní a následně se pomocí integrovaného editoru programuje jeho chování (MathWorks, 2016).

4.2 OPTIMIZE LAB

Optimize Lab je software, který je vyvíjen společností Stäubli. Slouží k analýze provozního stavu daného robota. Vyvíjen je v prostředí Matlab. Je to nástroj k optimalizaci běhu programu daného robota. K analýze se využívá záznam cyklu programu, který obsahuje všechny potřebné parametry (od souřadnic pohybu, úhlů natočení kloubů až po napětí na motorech a tlakového zatížení převodovky). Tyto parametry jsou zpracovány do grafů, které obsluha následně vyhodnotí. Výhodnou možností je použití v kombinaci s SRS studiem pro barevné vyznačení dráhy ramena, ve kterém lze spatřit místa, při kterých dochází k přetěžování součástí, které má za následek snížení doby životnosti robota.

5 KOMUNIKAČNÍ ROZHRANÍ

Cílem průmyslového komunikačního rozhraní je propojit všechny části provozu do jednoho systému, to zvyšuje dostupnost dat a jejich efektivitu. Tyto systémy lze rozdělit do tří vrstev. První je vrstva senzorů a akčních členů, která je vhodná pro komunikaci v reálném čase se senzory. Druhá vrstva se týká sběrniceových zařízení, která se používají pro komunikaci na úrovni programovatelných automatů. Zajišťuje propojování I/O zařízení a ovládacích zařízení. Třetí vrstva obsahuje průmyslové komunikační systémy, které umožňují provádět událostmi řízené služby (ElektroPrůmysl, 2013).

5.1 TYPY KOMUNIKACÍ

V dnešní době existuje spousta průmyslových rozhraní. Mezi celosvětově rozšířené protokoly komunikující v reálném čase patří například EtherCAT. Síť je postavena na principu master-slave a vychází z Ethernetu. Jeho výhodou jsou krátké doby cyklů, vysoký výkon, žádná potřeba speciálního hardwaru a přesná synchronizace (Automa, 2016).

Dalším zástupcem této komunikace je Ethernet Powerlink. Stejně jako EtherCAT, i Ethernet Powerlink vychází z protokolu Ethernet. Tento protokol nevyžaduje žádný speciální hardware, vystačí si s hardwarem určeným pro Ethernet. Dosahuje výborných vlastností při práci v reálném čase čistě softwarovým řešením (Zezulka, 2016).

Ethernet/IP je jeden z široce používaných průmyslových protokolů vycházejících z Ethernetu, který byl vyvinut pro průmyslovou automatizaci. Stejně jako Powerlink využívá standardní prostředky Ethernetu a nepožaduje tedy speciální software. Výhody této technologie je pohodlná konfigurace, snadný přístup k procesním proměnným, instrumentaci a diagnostickým údajům, a bezproblémová integrace do řídicích systémů pomocí přímého připojení přes sběrnici Ethernet/IP (Zezulka, 2016).

Dalšími známými zástupci je Profinet a Profibus.

5.1.1 Socket

Síťový socket je označení pro koncový bod připojení. Většina komunikací pak využívá protokoly TCP/IP, kde vlastní přenos zajišťuje IP protokol. Socket je charakterizován dvěma parametry, a to IP adresou a číslem portu, případně ještě protokolem (UDP nebo TCP). Socket pracující s protokolem TCP vyžaduje roli klienta a serveru.

Komunikace následně probíhá tak, že server očekává žádost u spojení. Po navázání komunikace klientem začne probíhat komunikace, dokud není ukončena (Vokřínek, 2017).

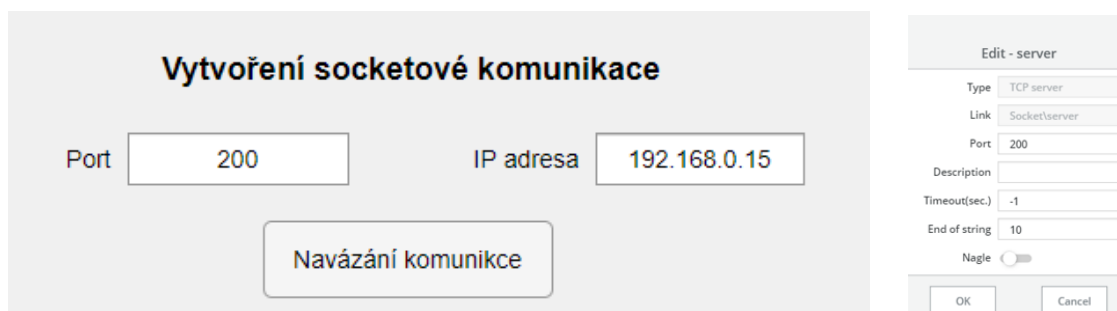
6 NÁVRH APLIKACE

Aplikace pro řízení robota Stäubli řady TX2 je vytvořena v prostředí Matlab za použití App Designeru, který umožňuje intuitivní vytvoření grafické aplikace. Jedná se o grafický nástroj pro tvorbu aplikací. Jednotlivé komponenty se vytvářejí tahem myši a následně probíhá automatické generování kódu pro vytvoření komponenty spolu s daty o umístění a velikosti.

Princip řízení je založen na posílání příkazů pomocí socketové komunikace mezi Matlabem a robotem. Funguje zde vztah klient-server. V pozici serveru je zde robot, klientem se tedy stává Matlab. Aplikace má za úkol posílat požadavky robotu, ať už jako pohybové úkony, nebo příkazy pro získání stavových informací. Robot má za úkol požadavek vyslyšet, zpracovat a provést dané úkony.

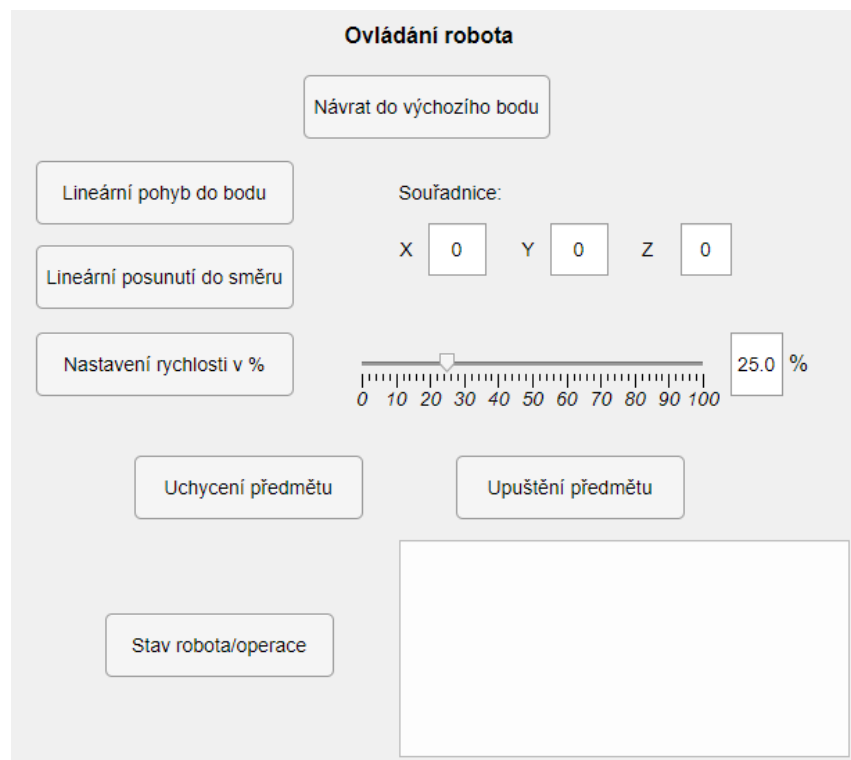
6.1 GRAFICKÉ ROZHRAŇÍ

Grafické rozhraní se skládá ze dvou částí. První část má za úkol vytvořit socketový kanál, přes který proudí veškerá data. K tomu je zapotřebí znát číslo portu a IP adresu robota. Číslo portu si určuje uživatel na straně robota při vytvoření serveru, viz Obrázek 6.1. IP adresu lze nalézt v nastavení na daném teach pendantu u robota.



Obrázek 6.1 – Tvorba kanálu, vytvoření serveru

Druhá část se zabývá vlastními požadavky. Veškerá obsluha je formou textových polí, posuvníku a tlačítek, která zajišťují odesílání požadavků, viz Obrázek 6.2.



Obrázek 6.2 – Ovládací aplikace

6.2 POPIS FUNKCÍ APLIKACE

První funkcí je „Návrat do výchozího bodu“. Jak z názvu vyplývá, jedná o se o návrat do předem definovaného bodu. Je doporučeno použít tuto funkci při každém prvním spuštění řízení robota, není to nicméně nutné, víme-li v jakém místě se robot nachází.

Další dvě funkce se týkají lineárního pohybu. První z nich určuje bod, do kterého se rameno přenesou. Bod je určen trojicí souřadnic x , y , z . Druhá funkce lineárně posune robotické rameno do určitého směru podle souřadnic x , y , z , případně jejich kombinací. Zadávané souřadnice jsou v jednotkách milimetrů, lze využít i desetinná čísla.

Další funkce se zabývá změnou rychlosti. Je možné využít posuvníku, případně textového pole pro přesnou hodnotu. Hodnota je udávána procentuálně vůči maximální rychlosti.

Následující funkce se zabývá nástrojem, konkrétně v tomto případě kleštěmi. Stiskem prvního tlačítka dojde k uchycení předmětu, druhým tlačítkem se předmět uvolní.

Poslední funkcí je získání informací o pozici ramene a nastavené rychlosti.

7 PROGRAMOVÉ ŘEŠENÍ

Programové řešení je rozdělené na dvě části. První část se zabývá kódem v prostředí Matlab. Obsahuje vytvoření socketu, příslušný kód pro grafickou stránku aplikace a jednotlivé požadavky na řízení. Druhá část se nachází v SRS studiu, kde se vyskytuje sledování komunikace a následné reakce na daný požadavek.

7.1 MATLAB

Veškeré programování probíhalo v App Designeru, kódy se tvořily ve volacích funkcích pro tlačítka, která po stisku ihned odesílají příkazy přes socketový kanál do robota.

7.1.1 Socket

Pro vytvoření socketu jsou důležité dva parametry, jedná se o IP adresu a číslo portu komunikace. Obě informace zadává uživatel do textových polí, viz Obrázek 6.2.

```
IP = app.IPEditField.Value;  
Port = app.PortEditField.Value;  
app.socket = tcpip(IP, Port, "NetworkRole", "client", "InputBufferSize", 150);
```

Dále je možné využít dalších parametrů. V tomto případě stačilo určit roli klienta a omezit vstupní velikost paměti. Zcela nezbytný je příkaz *fopen*, který vytváří socketový kanál, a to umožňuje zahájit komunikaci. Pro ukončení komunikace se naopak používá *fclose*.

```
fopen(app.socket);  
fclose(app.socket);
```

7.1.2 Další funkce

Pro zapisování dat na socket, a odesílání tak příkazů, se používá funkce *fwrite*.

```
fwrite(app.socket, "Speed;" + num2str(speed) + "#");
```

Parametry této funkce jsou používaný socket a textový řetězec, který se odesílá. Při použití funkce *num2str* dojde k převodu čísla na text, aby to bylo možné je odeslat současně spolu s textových řetězcem. Při použití funkce *fwrite* dochází k převodu textových řetězců do číselného ascii kódu, aby mohla být data úspěšně odeslána socketem. Zpětný převod na textový řetězec by mohl být komplikovaný, jelikož ze strany serveru nelze zjistit, kde zpráva

končí. Proto jsou použity speciální znaky. Středník je použit pro oddělení jednotlivých slov, zatímco křížek značí konec celého řetězce.

Pro čtení socketu lze využít funkci *scanf*.

```
fscanf(app.socket);
```

Tato funkce za nás převádí data ze socketu z ascii kódu na textový řetězec, tudíž není potřeba řešit žádný převod, nicméně je potřeba rozdělit zprávu na jednotlivé textové řetězce.

```
for i=1:150
    if zprava(i)~=char(59) && pom==0 && zprava(i)~=char(35)
        s1 = s1 + zprava(i);
    elseif zprava(i)==char(59)
        pom = pom + 1;
    elseif zprava(i)~=char(59) && pom==1 && zprava(i)~=char(35)
        s2 = s2 + zprava(i);
    elseif zprava(i)~=char(59) && pom==2 && zprava(i)~=char(35)
        s3 = s3 + zprava(i);
    elseif zprava(i)~=char(59) && pom==3 && zprava(i)~=char(35)
        s4 = s4 + zprava(i);
    elseif zprava(i)==char(35)
        state = [s1 ; " " ; s2 ; " " ; s3 ; " " ; s4];
        app.StateArea.Value = state;
        break;
    end
end
```

K tomu lze využít cyklus *for*. Ten prochází celý řetězec, který obsahuje 150 znaků, po jednom znaku a porovnává ho se speciálními znaky, které mezitím ukládá do další proměnné. Při nalezení shody se speciálním znakem se začnou následující znaky v pořadí ukládat do další proměnné do doby, než opět nadejde shoda se speciálním znakem. Posledním elementem v řetězci je speciální znak křížek, který indikuje konec příchozí zprávy. Následně se už jen vypíše vzniklé řetězce do textového pole pro zobrazení.

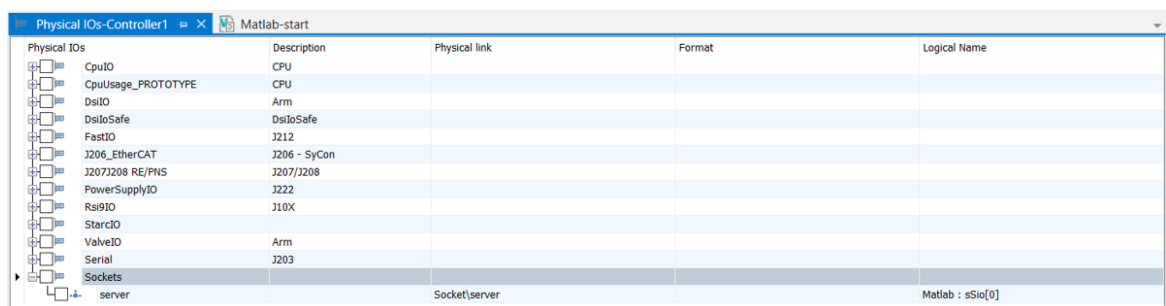
7.2 SRS STUDIO

Stäubli robotics suite studio (zkráceně SRS studio) je vývojové prostředí pro programování robotů Stäubli. Program v této sekci se zabývá hlídáním socketového kanálu a zpracováním požadavků.

Roboty Stäubli nemají implementovanou funkci pro automatické zpracování vnějších příkazů. Nelze tedy odeslat konkrétní příkaz z Matlabu pro pohyb či jiný úkon, ale je nutné vytvořit si vlastní funkce pro zpracování požadavku a následné reakce.

7.2.1 Socket

Vytvořit a nastavit server pro socketovou komunikaci lze dvojím způsobem. První způsob je vytvoření serveru na teach pendantu, viz Obrázek 6.2. Druhým způsobem je přidání fyzického vstupního portu v SRS studiu. Následně stačí přidělit proměnnou pro vytvořený kanál a tím dojde k vytvoření serveru.



Physical IOs	Description	Physical link	Format	Logical Name
CpuIO	CPU			
CpuUsage_PROTOTYPE	CPU			
DsiIO	Arm			
DsiIoSafe	DsiIoSafe			
FastIO	J212			
J206_EtherCAT	J206 - SyCon			
J207/J208 RE/PMS	J207/J208			
PowerSupplyIO	J222			
Rsi9IO	J10X			
StarCIO				
ValveIO	Arm			
Serial	J203			
Sockets				
server		Socket\server		Matlab : sSio[0]

Obrázek 7.1 – Socketový server v SRS studiu

Pro přečtení příchozí zprávy se využívá příkaz *sioGet*.

```
sioGet(sSio, nObsahSocketu)
```

Kde *sSio* je proměnná představující daný kanál a *nObsahSocketu* představuje proměnnou, do které se ukládá příchozí zpráva po jednom znaku. Příkaz je použitý v cyklu *while* pro neustálé sledování kanálu. Po přijetí zprávy je vhodné převést zprávu z číselného ascii kódu do textového řetězce. K tomu lze využít konstrukci *ifelse* a příkaz *chr*, který zajišťuje převod.

```

if nObsahSocketu !=59 and nPom==0
    sA=sA+chr(nObsahSocketu)
elseif nObsahSocketu==59
    nPom=nPom+1
elseif nObsahSocketu !=59 and nPom==1 and sA=="Speed"
    sSpeed=sSpeed+chr(nObsahSocketu)
elseif nObsahSocketu !=59 and nPom==1
    sX=sX+chr(nObsahSocketu)
elseif nObsahSocketu !=59 and nPom==2
    sY=sY+chr(nObsahSocketu)
elseif nObsahSocketu !=59 and nPom==3
    sZ=sZ+chr(nObsahSocketu)
endif

```

Pro zápis do socketu se využívá příkaz *sioSet*.

```
sioSet(sSio, nText)
```

Nicméně před samotným odesláním zprávy je nutný převod textových řetězců do ascii kódu. To lze provést pomocí cyklu *for* a příkazu *asc*.

```

for l_nIndex=0 to len(sText)-1
    nText[l_nIndex] = asc(sText,l_nIndex)
endFor

```

7.2.2 Pohybové funkce

Pro pohyb do výchozího bodu se využívá příkaz *movej*.

```

movej(jHome,tTool,mNomSpeed)
waitEndMove()

```

Jedná se o kloubový pohyb, jehož parametry jsou *jHome*, který představuje souřadnice, které jsou uživatelsky pevně nastavené v prostoru robota, *tTool* určuje, zda je přítomen nástroj, a případě upravuje souřadnice v závislosti na velikosti nástroje, a *mNomSpeed*, který udává rychlost pohybu. Příkaz *waitEndMove* zajišťuje, že pohyb bude dokončen až v daném bodě a nebude přerušeno následujícím příkazem.

Pro lineární pohyb existuje příkaz *movel*.

```

movel(pP,tTool,mNomSpeed)
waitEndMove()

```

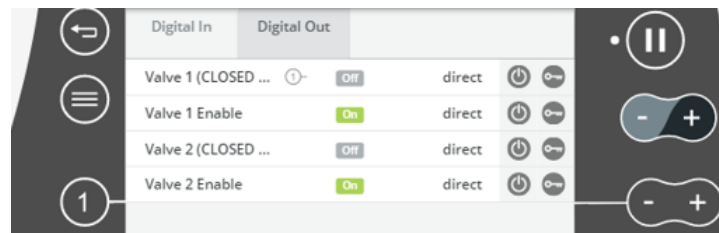
Pro použití nástroje se používá pár příkazů *close* pro uchycení a *open* pro upuštění předmětu.

```

close(tTool)
open(tTool)

```

Před samotným spuštěním programu je nutné povolit použití ventilů v nastavení na teach pendantu. Jedná se o bezpečnostní prvek.



Obrázek 7.2 – Povolení užívání ventilů

7.2.3 Ostatní funkce

Pro změnu rychlosti lze využít příkaz *toNum*.

```
toNum(sSpeed,mNomSpeed.vel,bB)
```

Tento příkaz převádí rychlost z textového formátu do číselného. Parametr *sSpeed* představuje text, který je zapotřebí převést. Do druhého parametru se ukládá převedené číslo a zároveň nastavuje rychlost na požadovanou hodnotu.

Pro vytvoření uživatelského rozhraní pro teach pendant lze využít příkaz *userPage*.

```
userPage("help")
```

Kde *help* představuje jméno vytvořeného rozhraní ve formátu *.html*. Pro vytvoření slouží intuitivní grafický designér implementován v SRS studiu.



Obrázek 7.3 – Uživatelské rozhraní

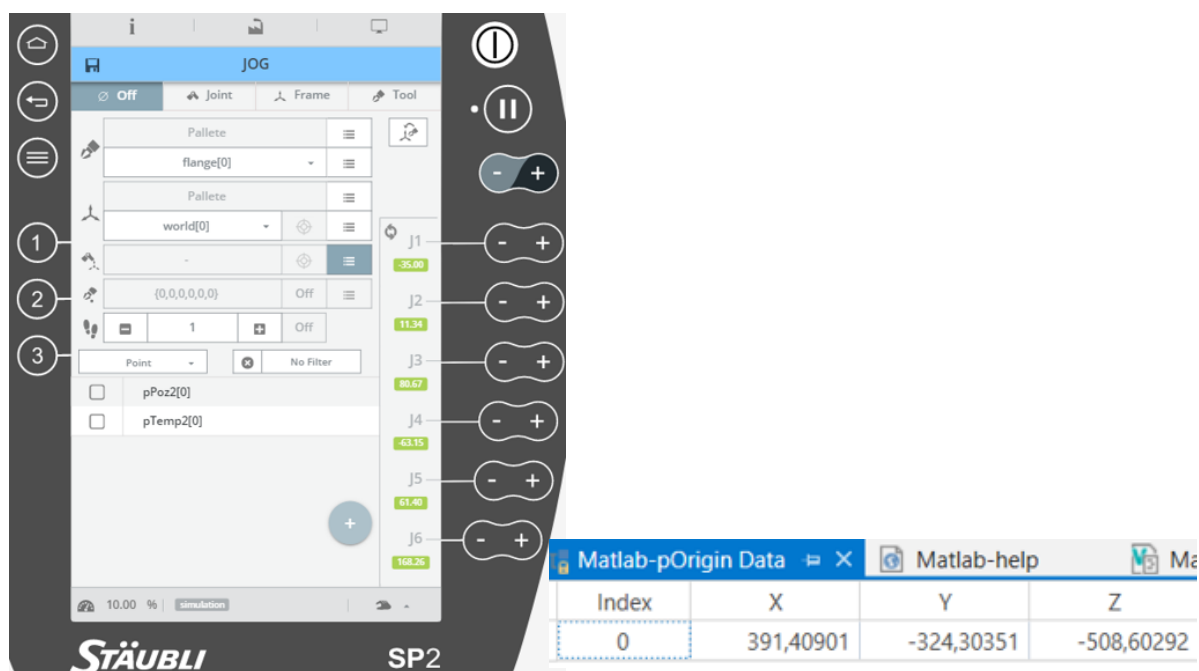
Posledním příkazem je *setFrame*, který nastavuje vlastní souřadnicový systém, tzv. rámec.

```
setFrame(pOrigin,pFrameOx,pFrameOxy,fFrame)
```

Pro vytvoření kartézského souřadnicového systému je potřeba zadat tři parametry. Jedná se o umístění nulového bodu, bodu na ose x a bodu na ose y. Čtvrtým parametrem v příkazu je jméno nově vzniklého rámce.

7.2.4 Určení bodů

Určit bodům souřadnice lze dvěma způsoby. Prvním je ruční zadání v SRS studiu. Efektivnější způsob je však určení bodů za pomoci teach pendantu umístěním robota do požadovaného místa a uložení dané pozice. Před samotným uložení je nutné vybrat bod, který je potřeba nastavit, používaný nástroj a rámec, do kterého daný bod spadá.

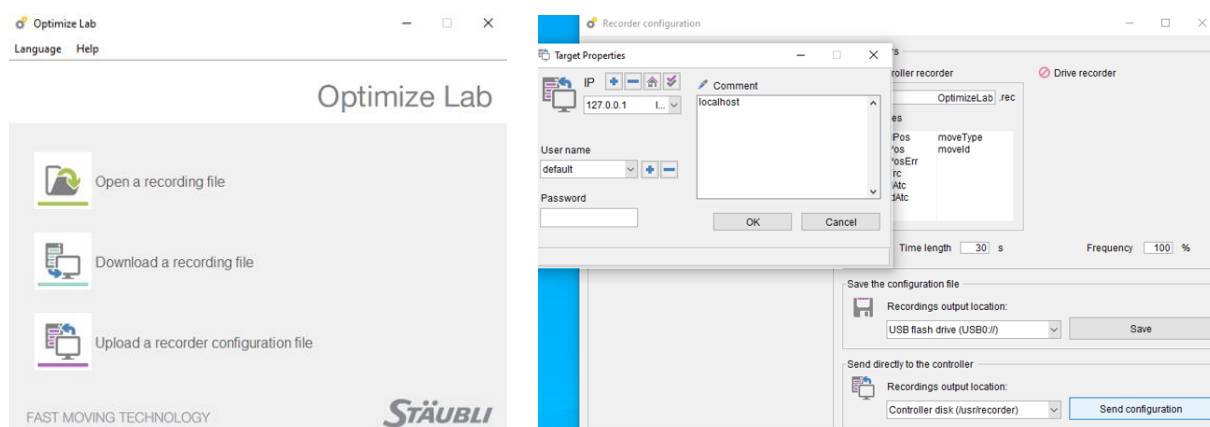


Obrázek 7.4 – Nastavení souřadnic bodů

8 OPTIMIZE LAB

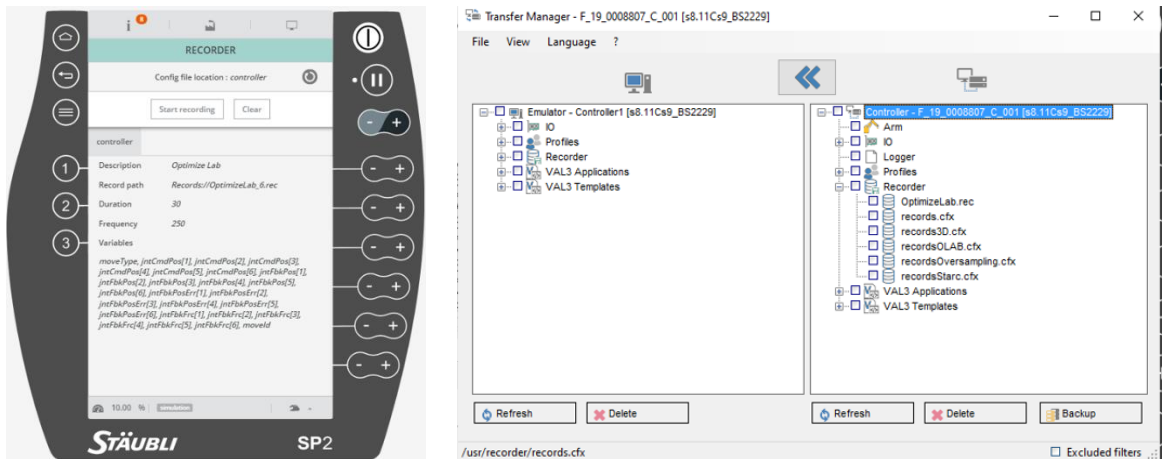
8.1 ZÁZNAM A KONFIGURACE

Záznamy cyklu lze načíst dvojím způsobem. První možnost je otevření již existujícího záznamu, který může být uložen na libovolném médiu. Druhá možnost je přímé stažení nahrávky z robota za předpokladu, že robot je správně nakonfigurován a obsahuje nahrávku. Při nahrávání konfiguračního souboru je potřeba znát IP adresu a používaný kontrolér. Následně je potřeba určit optimální délku cyklu, aby nedocházelo k zbytečnému zabírání paměti. A na závěr zadat účet uživatele, pod kterým dochází k práci. Typicky lze využít uživatele default.



Obrázek 8.1 – Úvodní okno, nastavení

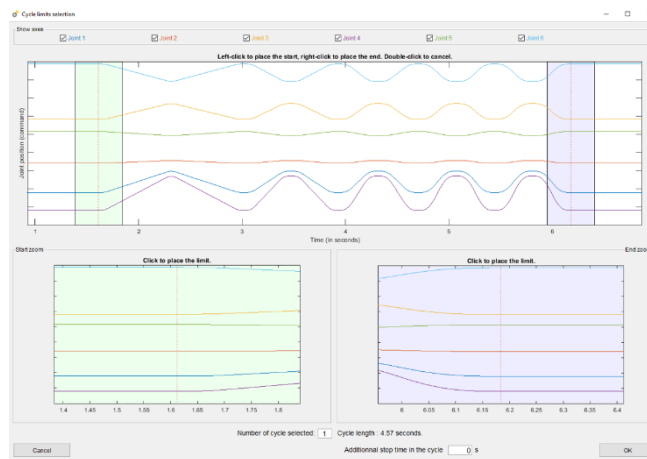
Při takto nakonfigurovaném robotu lze na teach pendantu spustit nahrávání záznamu. Mimo samotného spuštění je zde vidět název souboru, typicky ve formátu OptimizeLab_X.rec, kde X je další číslo v pořadí po předchozích záznamech. Staré záznamy lze smazat pomocí FTP serveru nebo v transfer manageru v SRS studiu.



Obrázek 8.2 – Spuštění záznamu, zobrazení v transfer manageru optimalizace

8.2 DÉLKA CYKLU A ŽIVOTNOST

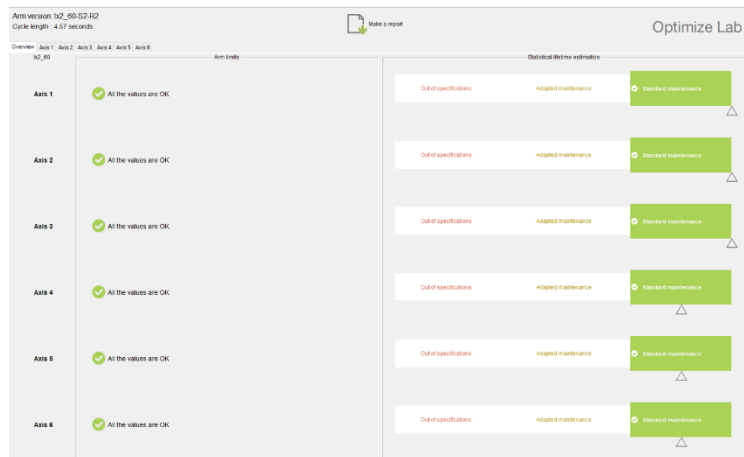
Po otevření záznamu se zobrazí první graf, na kterém lze spatřit pozice kloubů ramena po dobu celého cyklu. V této fázi optimalizace upřesníme velikost doby začátku a konce cyklu v případě, že záznam je zbytečně dlouhý, případně máme-li zájem o konkrétní část záznamu. Tu lze omezit prostým dvojklikem myši do oblasti grafu v požadovaný čas.



Obrázek 8.3 – Nastavení doby cyklu

Po potvrzení délky cyklu se zobrazí počet os robota v levé části. V pravé části se ke každé ose zobrazí velikost zatížení určená šipkou na spodní straně. Zatížení je rozdělené na tři části. Ideální stav je v zelené, která představuje bezproblémový chod a dlouhou životnost bez potřeby častějších kontrol. Ve žlutém poli dochází k mírnému zatížení a je doporučená

častější kontrola. Červená část představuje silné zatížení, dochází k významnému snížení životnosti a je doporučena optimalizace programu.

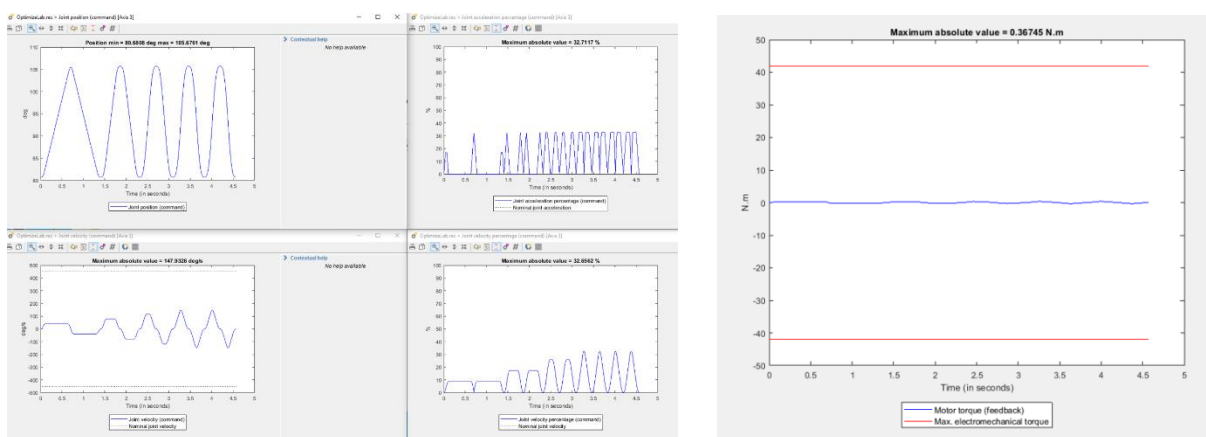


Obrázek 8.4 – Ukázka životnosti

V horní části lze vybrat konkrétní osu a na ní si zobrazit další parametry. Pro každou osu existují parametry pro klouby, motor a převodovku.

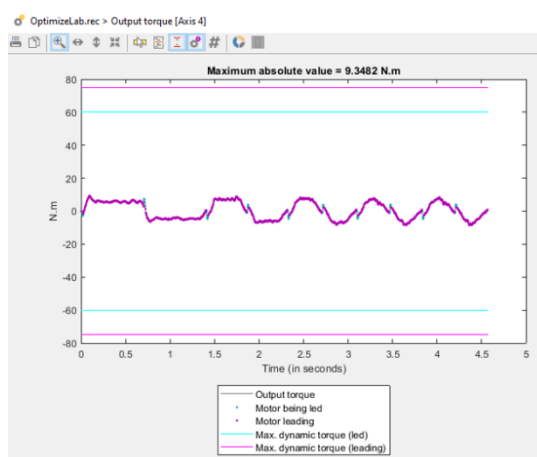
8.3 UKÁZKA PARAMETRŮ

Parametry pro klouby jsou pozice kloubu (natočení), zrychlení a rychlost (procentní vyjádření případně ve stupních za sekundu). Dále je možné zobrazit si žádané hodnoty případě zpětnovazební. Parametr pro motor je točivý moment.



Obrázek 8.5 – Zobrazitelné parametry

Nejdůležitějším parametrem je točivý moment převodovky.

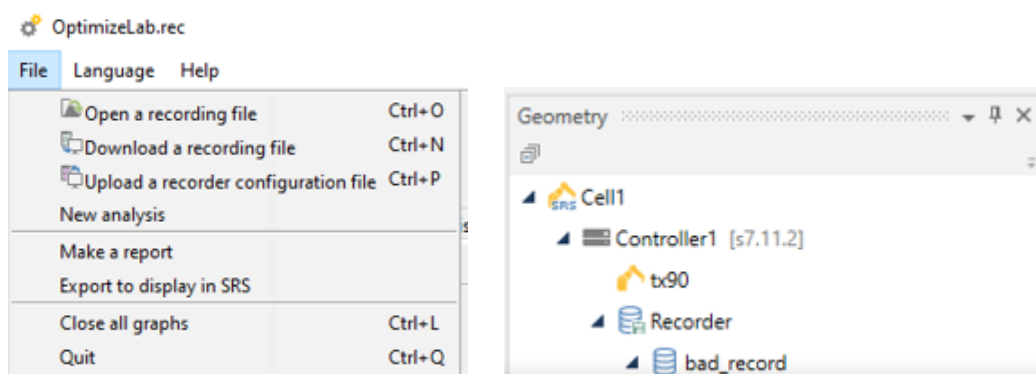


Obrázek 8.6 – Točivý moment převodovky

Téměř ve všech grafech jsou zobrazované hranice hodnot (nejčastěji červenou přímkou), které určují oblast, ve které nedochází k přetěžování dané součásti. Při překročení hranic lze výhodně využít simulace pro zobrazení barevné dráhy, na které lze jasně vidět, ve kterém místě dochází k zatížení.

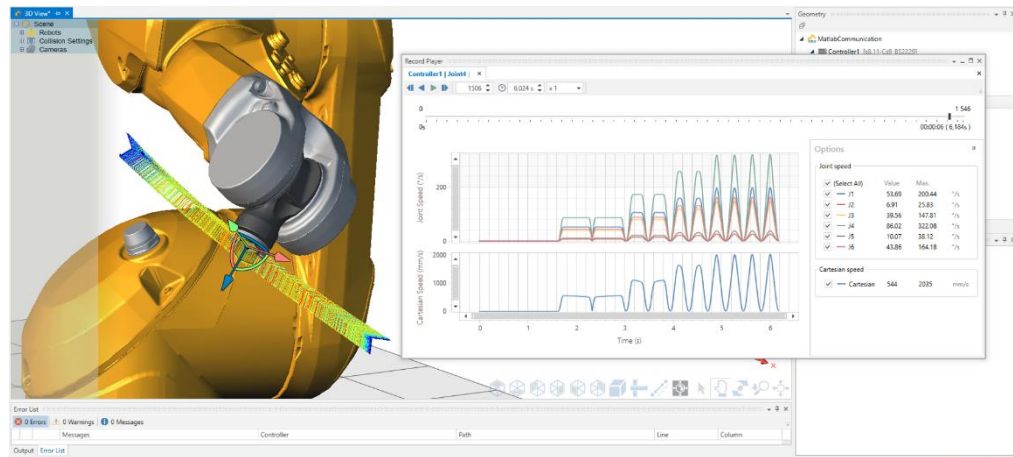
8.4 SIMULACE

Simulace se provádí v SRS studiu. Té lze dosáhnout exportováním dat z Optimize Labu. Provede se to možností „Export to display in SRS“ na záložce file. Následně stačí spustit SRS studio, kde se v záložce Geometry najde položka Recorder a pod ní název, který se zadal při exportování z Optimize Labu, dané simulace pro zobrazení barevně vyznačené dráhy.



Obrázek 8.7 – Export do SRS studia, umístění

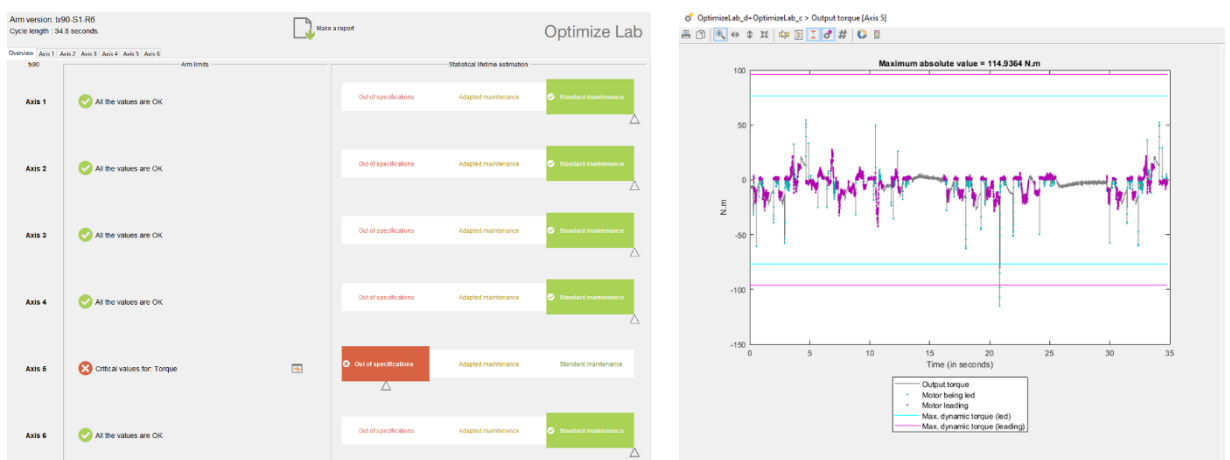
Po spuštění 3D zobrazení se zobrazí odpovídající robotické rameno s barevně vyznačenou dráhou, případně i se zobrazením grafů rychlosti otáčení kloubů. Reálný pohyb ramene si lze zobrazit spuštěním simulace, díky kterému je možné zjistit přesný pohyb všech os. V případě použití nějakého nástroje je možné si ho namodelovat a taktéž použít při simulaci pro věrné zobrazení.



Obrázek 8.8 – Simulace pohybu

8.5 UKÁZKA ŠPATNÉ OPTIMALIZACE A MOŽNÉ ŘEŠENÍ

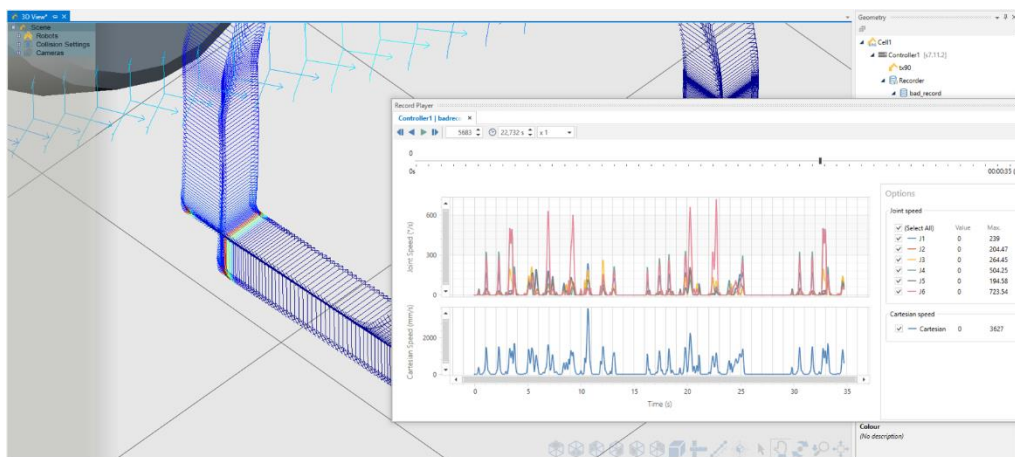
Při špatné optimalizaci pohybu Optimize Lab ukáže, ve které ose dochází k nějakému přetěžování a odkáže uživatele na daný problém. Na Obrázek 8.9 lze vidět, že v jeden moment dochází k překročení limitní hranice pro točivý moment převodovky.



Obrázek 8.9 – Vzor zatížení v Optimize Lab

Po spuštění 3D zobrazení v SRS studiu lze vidět červeně zabarvenou část, ve které dochází k přetížení, viz Obrázek 8.10. Při spuštění simulovaného pohybu ramene by se zjistilo, že v daném místě dochází k chvilkovému pozastavení za účelem manipulace s produktem, s následným pokračováním pohybu. Z toho lze vyvodit závěr, že k přetížení převodovky dochází při prudkém zrychlení.

Tento problém je možné vyřešit dvojím způsobem. První možností je vyvarovat se zastavení, je-li to možné. Dochází-li v daném místě otočení kloubu kvůli používanému nástroji, tak je možné natočit kloub během předchozího pohybu, a tudíž se tak vyvarovat zastavení. Druhou možností je snížit velikost zrychlení a díky tomu snížit námahu používané převodovky.



Obrázek 8.10 – Vzor zatížení v simulaci

8.6 VYUŽITÍ ADDONŮ

V případě potřeby pravidelného automatického zaznamenávání pohybu lze využít recorder addon, který lze najít v technické databázi Stäubli. Umožňuje programově zajistit opakované nahrávání záznamu drah bez nutnosti jakéhokoli vnějšího zásahu od uživatele.

9 ZÁVĚR

Tato práce se zaměřuje na prostředí Matlab a jeho možnosti využití pro řízení robota Stäubli TX2 a optimalizaci těchto robotů za pomoci nástroje Optimize Lab.

V teoretické části práce jsou popsány roboty a jejich využití. Detailnější pohled je na robota Stäubli z řady TX2, u které jsou představeny její přednosti, protože byl použit pro praktickou ukázkou. Také se tato část zabývá prostředími, jako je SRS studio, které slouží pro programování robotů Stäubli, ale také Matlabem. Zároveň se zabývá i aktuálně používanými průmyslovými komunikacemi a síťovým socketem, který je využit v praktické části pro komunikaci.

V praktické části byla navržena řídicí aplikace v prostředí Matlab, která zajišťuje komunikaci s SRS studiem pomocí vytvořeného socketu.

V první části byla detailně popsána grafická podoba aplikace spolu s jejími funkcemi. Jako funkční řešení pro řízení robotů bylo zvoleno využití socketové komunikace pod protokolem TCP/IP mezi Matlabem a robotem. V druhé části byl kladen důraz na samotné programování, které se zabývalo jak správným rozklíčováním posílaných zpráv přes socket, tak zajištěním odpovídajících reakcí. Nevýhodou robotů Stäubli je neumožnění zpracování přímých externích příkazů, a bylo tedy nutné vytvořit vlastní programové funkce, které řešily reakce na příchozí požadavky.

Výsledkem praktické práce byla funkční aplikace pro řízení robotů Stäubli z řad TX2 v prostředí Matlab, která zajišťuje pohyb ramene, používání nástrojů, získávání stavových zpráv a řízení rychlosti pohybu. Součástí řešení bylo i vytvoření druhého programu pro zpracování příchozího požadavku a následnou reakci na straně robota.

Na závěr je otestován nástroj Optimize Lab vyvíjen společností Stäubli. Nachází se zde detailně popsané možnosti získání záznamu a následně jeho zpracování. Byly ukázány důležité indikační parametry a zároveň byla představena možnost využití Optimize Lab v kombinaci s SRS studiem pro zobrazení dráhy robota a simulaci jeho pohybu. Díky tomuto programu je možné najít kritická místa v dráze pohybu a předejít tak zatěžování převodovky, což by mohlo způsobit snížení životnosti robota.

POUŽITÁ LITERATURA

ABB YuMi [foto]. *ABB* [online]. [cit. 2021-7-26]. Dostupné z:

<https://new.abb.com/products/robotics/cs/kolaborativni-roboty/yumi>

App Designer. *MathWorks* [online]. 2016 [cit. 2021-7-26]. Dostupné z:

<https://www.mathworks.com/products/matlab/app-designer.html>

BLAŠKA, J., KRUMPHOLC, M., SEDLÁČEK, M.: Využití grafického uživatelského rozhraní Matlabu ve výzkumu a výuce měření. *Konference Matlab* [online]. ČVUT, Praha, 2008, s. 1-6. [cit. 2021-7-26]. Dostupné z:

http://dsp.vscht.cz/konference_matlab/matlab03/blaska.pdf

DELTA [foto]. *AutoCont Control Systems* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.accs.cz/produkty-mitsubishi-electric/roboty/delta>

Driving Stäubli robots with industrial PLCs. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.staubli.com/en-mx/robotics/product-range/robot-software/val3-robot-programming/unival-solutions/unival-plc/>

EPSON SCARA LS10-B702S [foto]. *Epson* [online]. Seiko Epson Corporation [cit. 2021-7-26]. Dostupné z: <https://www.epson.cz/products/robot/epson-scara-ls10-b702s-standardni>

EtherCAT Automation Protocol. *Automa* [online]. 2016 [cit. 2021-7-26]. Dostupné z:

https://automa.cz/cz/casopis-clanky/ethercat-automation-protocol-2017_02_0_9811/

HAVLÍČEK, Daniel. Co jsou to SCARA roboty? *Factory Automation* [online]. FANUC Czech, 23. 03. 2018 [cit. 2021-7-26]. Dostupné z: <https://factoryautomation.cz/co-jsou-to-scara-roboty/>

HelMo the mobile robot system. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.staubli.com/en/robotics/product-range/helmo-mobile-robot-system/>

Inovativní mechatronická řešení pro vyšší produktivitu. *Stäubli* [online]. [cit. 2021-7-26].

Dostupné z: <https://www.staubli.com/cs-cz/>

MATLAB & SIMULINK. *HUMUSOFT* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.humusoft.cz/matlab/details/>

NJ-420-3.0 [foto]. *Berger Positec* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.regulacni-pohony.cz/inpage/robustni-do-650-kg/>

Our 6 axis industrial robots. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.staubli.com/en/robotics/product-range/industrial-robots/6-axis-robots/>

- Our SCARA industrial robots. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en/robotics/product-range/industrial-robots/4-axis-scara-robots/>
- Průmyslové komunikační sítě. *ElektroPrůmysl* [online]. 27. 01. 2013 [cit. 2021-7-26]. ISSN 2571-0761. Dostupné z:
<https://www.elektroprumysl.cz/automatizace/prumyslove-komunikacni-site>
- Rozdíly mezi kartežskými, šestiosými a SCARA roboty. *E-konstruktor* [online]. Nová média, 14. 04. 2014 [cit. 2021-7-26]. Dostupné z:
<https://e-konstrukter.cz/novinka/rozdilky-mezi-kartezskeymi-sestiosymi-a-scara-roboty>
- SCARA [foto]. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en/robotics/product-range/industrial-robots/4-axis-scara-robots/>
- Stäubli představuje tři nové modely TX2 6-osých robotů. *Plastic Portal* [online]. ICOSA, 11. 02. 2021 [cit. 2021-7-26]. Dostupné z: <https://www.plasticportal.cz/cs/staubli-predstavuje-tri-nove-modely-tx2-6-osych-robotu.html/c/7131/>
- Stäubli Robotics Suite. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en-us/robotics/product-range/robot-software/pc-robot-programming-srs/>
- Svářecí robot [foto]. *Ferier* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.ferier.cz/svarovaci-roboty-cloos/system-robotizace-mechanika/>
- TX2 [foto]. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en/robotics/product-range/industrial-robots/6-axis-robots/>
- TX2touch POWER cobot range. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en-mx/robotics/product-range/cobots/power-cobot/>
- TX2 robot range [online]. 2021 [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en/file/21193.show>
- VAL 3 language. *Stäubli* [online]. [cit. 2021-7-26]. Dostupné z:
<https://www.staubli.com/en-mx/robotics/product-range/robot-software/val3-robot-programming/val-3-language/>
- VOKŘÍNEK, Jiří. Sockety a síťování. *Prezentace* [online]. Praha: ČVUT, 2017 [cit. 2021-7-26]. Dostupné z:
<https://cw.fel.cvut.cz/old/media/courses/b0b36pju/prednasky/lecture09-handout-2x2.pdf>

We move your business. *Stäubli WFT* [online]. [cit. 2021-7-26]. Dostupné z:

<https://www.staubli-wft.com>

ZBÍRALOVÁ, Kristýna. Co je to Delta robot a jak funguje? *Factory Automation* [online].

FANUC Czech, 15. 02. 2019 [cit. 2021-7-26]. Dostupné z:

<https://factoryautomation.cz/delta-robot-jak-funguje/>

ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet VIII: Ethernet

Powerlink, Profinet. *Automa* [online]. 2016 [cit. 2021-7-26]. Dostupné z:

https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-viii-ethernet-powerlink-profinet-2008_05_37288_6341/

ZEZULKA, František a Ondřej HYNČICA. Průmyslový Ethernet IX: EtherNet/IP,

EtherCAT. *Automa* [online]. 2016 [cit. 2021-7-26]. Dostupné z:

https://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-ix-ethernet/ip-ethercat-2008_10_37910_6510/

ŽÁČEK, Michal. Průmyslové roboty: Jaké jsou jejich druhy? *Factory Automation* [online].

FANUC Czech, 23. 03. 2018 [cit. 2021-7-26]. Dostupné z:

<https://factoryautomation.cz/prumyslove-roboty-jake-jsou-jejich-druhy/>

PŘÍLOHY

A – CD

Příloha k bakalářské práci

MOŽNOSTI PROSTŘEDÍ MATLAB PRO OPTIMALIZACI A PROGRAMOVÁNÍ
ROBOTŮ STÄUBLI

Jan Fridrich

CD

OBSAH

1. Text bakalářské práce ve formátu PDF
2. Úplný zdrojový kód aplikace v prostředí Matlab
3. Úplný zdrojový kód aplikace v prostředí SRS