

Řídicí systém stroje pro broušení plošných materiálů

Diplomová práce

Studijní program:

N2612 Elektrotechnika a informatika

Studijní obor:

Mechatronika

Autor práce:

Bc. Tomáš Rompotl

Vedoucí práce:

doc. Dr. Ing. Jaroslav Hlava

Ústav mechatroniky a technické informatiky





Zadání diplomové práce

Řídicí systém stroje pro broušení plošných materiálů

Jméno a příjmení: **Bc. Tomáš Rompotl**
Osobní číslo: M18000168
Studijní program: N2612 Elektrotechnika a informatika
Studijní obor: Mechatronika
Zadávací katedra: Ústav mechatroniky a technické informatiky
Akademický rok: 2020/2021

Zásady pro vypracování:

Navrhněte řídicí systém stroje pro broušení plošných materiálů s rozměry asi 3x2 m. Systém musí být schopen provádět automatickou detekci polohy děr v plechu a regulovat přítlak tak, aby byly neustále zajištěny optimální brusné podmínky. Postupujte v následujících krocích:

1. Navrhněte algoritmy pro automatickou detekci polohy děr v plechu, regulaci přítlaku a další potřebné funkce řídicího systému. Bude-li to účelné, použijte simulaci pro ověření vlastností a funkčnosti algoritmů.
2. Analyzujte, jaké nároky budou při implementaci těchto algoritmů kladeny na pohony, sensoriku, PLC a další komponenty řízení a elektrických systémů stroje, a zvolte vhodné pohony a další související komponenty, které těmto nárokům vyhoví.
3. Algoritmy doplňte o funkce nutné k zajištění bezpečného provozu stroje.
4. Navrhněte hardware řídicího systému (PLC) a jeho propojení s pohony, sensorikou a dalšími elektrickými systémy stroje.
5. Algoritmy včetně bezpečnostních funkcí naprogramujte do zvoleného PLC a proveďte odzkoušení funkceschopnosti celého návrhu.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
40–50
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] W. Brian Rowe, Principles of Modern Grinding Technology, Elsevier 2014, ISBN 978-0-323-24271-4.
- [2] Lishu Lv, Zhaohui Deng, Tao Liu, Zhongyang Li, Wei Liu, Intelligent technology in grinding process driven by data: A review, Journal of Manufacturing Processes, Volume 58, 2020, pp. 1039-1051, ISSN 1526-6125.
- [3] Qilong Wang, Wei Wang, Lianyu Zheng, Chao Yun, Force control-based vibration suppression in robotic grinding of large thin-wall shells, Robotics and Computer-Integrated Manufacturing, Volume 67, 2021, 102031, ISSN 0736-5845.
- [4] C. H. Liu, Adrian Chen, C.-C.A. Chen, Yin-Tien Wang, Grinding force control in an automatic surface finishing system, Journal of Materials Processing Technology, Volume 170, Issues 1–2, 2005, pp. 367-373, ISSN 0924-0136,
- [5] P. Bitsindou, P. Guillaume, G. Delaunay, G. Villermain-Lecolier, Detection of holes, rivets or screws fixed on a metal sheet using a flat magnetic sensor, Sensors and Actuators A: Physical, Volume 24, Issue 3, 1990, pp. 181-186, ISSN 0924-4247.

Vedoucí práce:

doc. Dr. Ing. Jaroslav Hlava
Ústav mechatroniky a technické informatiky

Datum zadání práce:

9. října 2020

Předpokládaný termín odevzdání:

17. května 2021

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

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

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

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci 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.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

13. května 2021

Bc. Tomáš Rompotl

Poděkování

Tímto bych chtěl poděkovat všem, který se podíleli nebo mi pomohli realizovat tento projekt. Zejména panu Jaroslavu Hlavovi za užitečné a věcné rady a připomínky, Michalovi Čermákovi za možnost realizovat tento projekt a za technické konzultace. Dále Beckhoff Automation s.r.o. za pomoc při přípravě programu PLC.

Dále pak svým rodičům a slečně za podporu při psaní práce.

Řídicí systém stroje pro broušení plošných materiálů

Abstrakt

Práce se zabývá řídicím systémem stroje pro broušení plechů pásovou bruskou. Je zde probrána teorie broušení, představen broušený materiál a použitá bruska. Plechy obsahují díry a výřezy, což musí být bráno v úvahu při návrhu řídicího systému, aby bylo možné obrousit plech bez poničení hran. Na základě tohoto je diskutován algoritmus broušení a přistoupeno k návrhu algoritmu mapujícího obrys plechu. Algoritmus pracuje na principu nalezení a sledování hrany. S jeho pomocí stroj najde hrany plechu, a na základě těchto dat je poté přizpůsoben cyklus samotného broušení tak, aby hrany nebyly poničeny. Je provedena rešerše na podobné existující stroje, diskutováno fyzické uspořádání stroje a vybrány lineární pohony pro sestavení kartézského planárního manipulátoru. Součástí práce je i návrh elektroniky řídicího systému. Zde bylo jako řídicí jednotka vybráno IPC na platformě TwinCAT. Dále bylo navrženo elektrotechnické a pneumatické schéma zapojení. Práce je zakončena vývojem programu pro řídicí jednotku a implementace algoritmu do ní. Funkčnost řídicího systému byla odzkoušena na simulačním modelu stroje.

Klíčová slova: broušení, mapování obrysu, detekce hrany, Festo ELGA drive, Beckhoff TwinCAT, HIL

Control system of a metal sheet grinding machine

Abstract

This thesis deals with the control system of a machine for sheet metal grinding using a belt grinder. The theory of grinding is discussed, the grinded material and the used grinder are introduced. Sheets contain holes and cutouts, which must be considered when designing the control system so that the edges are not damaged while grinding. Based on this, the grinding algorithm is discussed and the algorithm for mapping sheet outline is designed. The algorithm is based on finding and tracking the edge. With the help of the developed algorithm, the machine finds the outlines of the sheet and based on this data, the grinding cycle itself is then adapted so that the edges are not damaged. A search for similar existing machines is realized, the physical arrangement of the machine is discussed and linear drives for the construction of a cartesian planar manipulator are selected. Thesis also deals with the design of control system electronics. The IPC on the TwinCAT platform was selected as the control unit. Then an electrical and pneumatic wiring scheme was designed. The thesis finishes with the development of the program for the control unit and the implementation of the algorithm in it. Functionality of the control system as whole was verified on the simulation model of the machine.

Keywords: grinding, outline mapping, edge detection, Festo ELGA drive, Beckhoff TwinCAT, HIL

Obsah

Seznam zkratk	13
Seznam značení veličin	14
Úvod	15
1 Teoretická část	16
1.1 Teorie broušení	16
1.1.1 Kinematika rovinného broušení	16
1.2 Upravovaný materiál	17
2 Bruska	19
3 Detekce obrysu plechu	20
3.1 Snímače pro detekci obrysu plechu	20
3.2 Plánování trasy	21
3.3 Algoritmus detekce hran	21
3.3.1 Detekce směru hrany	21
3.3.2 Určení směru pohybu snímačů	22
3.4 Mapování obrysu	23
3.4.1 Nalezení první hrany	24
3.4.2 Nalezení vnějšího obrysu plechu	24
3.4.3 Hledání děr v plechu	24
3.4.4 Bod v polygonu	24
3.5 Rozmístění snímačů	25
3.6 Ověření algoritmu hledání hran a plánování trasy simulací	26
3.6.1 Pomocné funkce	26
3.6.2 Nalezení první hrany	26
3.6.3 Nalezení vnějšího obrysu plechu	27
3.6.4 Zjednodušení hrany	28
3.6.5 Broušení a hledání děr	29
3.6.6 Průběh simulace	30
3.6.7 Ověření funkčnosti algoritmu hromadnou simulací	32
4 Mechanické řešení stroje	34
4.1 Analýza mechanických nároků na stroj	34
4.2 Rešerše existujících řešení	34
4.2.1 ZTPolish ZTD	34

4.2.2	AUTOPULIT CDA-2000x12000/2UP-CP	35
4.2.3	C.O.S.M.A.P. str1 - Surface grinding machine BM	35
4.2.4	Shrnutí řešerše	36
4.3	Původní studie zadavatele	36
4.4	Řešení postavené na lineárním vedení Festo ELGA	36
5	Pohony	38
5.1	Dimenzování pohonu osy X	38
5.2	Dimenzování pohonu osy Y	39
6	Elektronická část	41
6.1	Řídicí systém	41
6.1.1	Požadavky	41
6.1.2	Porovnání řídicích systémů	41
6.1.3	Zvolený řídicí systém	41
6.2	Vstupy a výstupy	42
6.3	Logická část	43
6.3.1	Konfigurace logické části	43
6.3.2	Bezpečnostní obvody	43
6.3.3	Dimenzování zdroje 24V	44
6.4	Silová část	45
6.4.1	Bruska a odsávání	45
6.4.2	Pohony os	45
6.4.3	Dimenzování přívodu	45
7	Pneumatická část	46
8	Řídicí systém	48
8.1	Program PLC systému TwinCAT	48
8.1.1	PLC runtime TwinCAT a ADS router	48
8.1.2	Struktura programu PLC	49
8.1.3	Řízení servopohonů os	49
8.2	HMI - TwinCAT TF2000	49
8.2.1	Obrazovka Přehled	50
8.2.2	Obrazovky Manuál a Přehled stroje	51
8.2.3	Obrazovka Nastavení	51
8.2.4	Obrazovky Info a Alarmy	51
8.3	Zpracování dat s využitím Matlab	52
8.3.1	Převod na spojitý pohyb	52
8.3.2	Komunikace s PLC	52
8.3.3	Kompilace na samostatnou aplikaci	53
8.4	Schéma systému jako celku	53
8.5	Simulace HIL	54
9	Závěr	55

Seznam obrázků

1.1	Ukázka broušeného materiálu 1	17
1.2	Ukázka broušeného materiálu 2	18
2.1	Pásová bruska GRIT GX 75	19
3.1	Detekce hrany plechu maticí čtyř snímačů, plech znázorněn šedou barvou	22
3.2	Postup určení tečny hrany - 2 senzory nad plechem	22
3.3	Postup určení tečny hrany - 1 senzor nad plechem	23
3.4	Postup určení směru pohybu (oranžová je směr pohybu)	23
3.5	Princip algoritmů určení přítomnosti bodů v polygonu [4]	25
3.6	Zadání přibližných souřadnic a nalezení hrany	30
3.7	Nalezení obrysu	31
3.8	Broušení	31
3.9	Vzorek výsledků hromadného testování	33
4.1	ZTPolish ZTD	35
4.2	AUTOPULIT CDA-2000x12000/2UP-CP	35
4.3	C.O.S.M.A.P. strl - Surface grinding machine BM	36
4.4	Původní návrh konstrukce brusky	37
4.5	Lineární pohon pohon Festo ELGA	37
5.1	Návrh Festo portálového manipulátoru	38
5.2	Výsledný profil pohybu osy X	39
5.3	Výsledný profil pohybu osy Y	40
6.1	Přehled uspořádání hlavních elektronických komponent	44
7.1	Schéma pneumatického obvodu	47
8.1	Panel řízení servopohonu podle PLCopen	51
8.2	Obrazovka Nastavení	52
8.3	Uspořádání řídicího systému	54

Seznam tabulek

2.1	Parametry pásové brusky	19
4.1	Mechanické nároky na stroj	34
5.1	Souhrn požadavků na osu X	39
5.2	Parametry zvoleného pohonu osy X	39
5.3	Souhrn požadavků na osu Y	40
5.4	Parametry zvoleného pohonu osy Y	40
6.1	Porovnání řídicích systémů	42
6.2	Souhrn vstupů	43
6.3	Souhrn výstupů	43
6.4	Spotřeba zařízení 24 VDC	44
6.5	Spotřeba zařízení napájených síťovým napětím	45

Seznam zkratek

Acc/Dcc Acceleration/Deceleration

COM Center of Mass

FB Funkční blok

HIL Hardware in the Loop

MC Move Control

NC Numerical Control

P2P, PtP Point to Point

PIP Point in Polygon

TUL Technická univerzita v Liberci

AI Analog Input

DI Digital Input

DO Digital Output

EtherCAT Ethernet for Control Automation Technology

ETC EtherCAT

IO, I/O Input/Output

SI Safety Digital Input

STO Safe Torque Off

USB Universal Serial Bus

ADS Automation Device Specification

B&R Bernecker & Rainer

CNC Computerized Numerical Control

FM Frekvenční měnič

HMI Human Machine Interface

HW Hardware

IoT Internet of Things

IP Internet Protocol

IPC Industrial PC

OS Operační systém

PC Personal Computer

PLC Programmable Logic Controller

IDE Integrated Development Environment

JSON JavaScript Object Notation

OOP Object Oriented Programming

OOIP Object Oriented Industrial Programming

TIA Portal Totally Integrated Automation Portal

TwinCAT The Windows Control and Automation Technology

TwinCAT XAE TwinCAT eXtended Automation Engineering

VCS Version Control System

VSCoDe Visual Studio Code

WYSIWYG What You See Is What You Get

XML eXtensible Markup Language

Seznam použitého značení

symbol	jednotka	popis veličiny
d_s	mm	průměr broušícího kotouče/bubnu
v_c	m/s	řezná rychlost
n_s	ot/min	rychlost otáčení brusky
r	1	rozlišení detekce směru
n	1	počet snímačů
m	1	váha stavu snímačů
s_i	bit	stav snímače i hlavy brusky
x_i, y_i	mm	poloha bodu
COM	[mm, mm]	poloha těžiště stavu snímačů
I_N	A	nominální fázový proud
k_u	1	koeficient současného využití
P_{Sp}	W	spotřeba zařízení napájených 24 VDC
P_{DC}	W	výkon zdroje 24 VDC

Úvod

Práce se zabývá návrhem a realizací řídicí části stroje pro automatické broušení nerezových plechů o maximálních rozměrech přibližně 3 x 2 m. Při obrábění by zároveň mělo být dosahováno optimálních brusných podmínek. Cílem je navrhnout řídicí systém schopný na základě přibližných souřadnic zadaných operátorem samostatně nalézt obrys plechu a obrousit celý jeho povrch. Tento algoritmus by bylo vhodné ověřit simulací. Součástí návrhu řídicího systému je i analýza nároků, vytipování fyzických komponent použitých na zařízení a vytvoření příslušných schémat. Jedná o řídicí jednotku, aktuátory a senzorku. Stroj by měl vyhovovat platným normám, a proto by měl splňovat i bezpečnostní požadavky. Algoritmus by měl být naprogramován do řídicího systému a odzkoušen na reálném zařízení.

V práci je probrán teoretický základ obrábění broušením a dále je čtenář seznámen s broušeným materiálem a použitou bruskou. Hlavní částí textu je kapitola zabývající se návrhem algoritmu pro strojové obroušení plechu tak, aby byl obroušen celý plech a zároveň nebyly zničeny hrany. Zde je postupně probrán navržený algoritmus zmapování obrysu plechu a broušení. Algoritmus je vyvíjen a jeho funkčnost ověřována za pomoci simulace v prostředí Matlab. Vedle teorie návrhu se práce zabývá analýzou nároků na řídicí systém a pohony a následným návrhem elektronického schématu a schématu pneumatiky. Nakonec je naprogramován řídicí systém.

Projekt je vyvíjen ve spolupráci firem Cermitech spol. s r.o. a Topos a.s. Ve druhé jmenované bude výsledný systém nasazen do testovacího provozu. Firma se zabývá výrobou strojních zařízení pro pekárenský průmysl. Tyto stroje jsou konstruovány z nerezové oceli. Nerezové plechy pro výrobu těchto strojů bývají uloženy ve skladu, kde povrchově degradují a vlivem způsobu uložení mohou být mírně deformovány. V případě použití tohoto materiálu pro výrobu stroje musí být odstraněny povrchové vady plechu - obroušení koroze a znečištění. Plechy jsou skladovány již obrobené s dírami a výřezy ve své ploše. Tyto charakteristiky musí být při broušení plechu brány v úvahu, při nevhodném přebroušení totiž hrozí zničení (stržení) hran. V současnosti je situace řešena manuální manipulací brusky operátorem. Použitá bruska má však pro manuální manipulaci vysokou hmotnost a výsledek mívá drobné vady. Motivací pro tuto práci bylo ulehčit pracovníkům broušení plechů a dosažení vyšší kvality obroušení. Stroj je z obou stran považován za prototyp/výzkumný kus a s ohledem na toto se odvíjí i způsob řešení některých problematik.

Projekt zároveň slouží jako testovací platforma, kdy bude pro řízení použit systém TwinCAT společnosti Beckhoff Automation s.r.o., což je přednostní systém používaný společností Cermitech, spol. s r.o. a bude ověřována využitelnost nového TwinCAT HMI a využitelnost verzovacího systému Git při vývoji.

1 Teoretická část

1.1 Teorie broušení

Úprava materiálu broušením se nejčastěji volí jako dokončovací metoda úpravy, a to pro získání hladkého povrchu nebo zbavení povrchu nežádoucí vrstvy jako oxidace nebo znečištění. Broušení se řadí mezi abrazivní metody obrábění, které jsou charakteristické použitím nástrojů s nedefinovatelnou řeznou geometrií břitu. Broušením lze obrábět i velmi tvrdé materiály.

Odebírání materiálu probíhá zrny nepravidelně rozmístěnými v brousicím nástroji. Zrna jsou malá, řádově zlomek milimetru a méně (záleží na hrubosti nástroje) a jsou tak schopná odebírat pouze malé množství materiálu. Navíc pojivo zrn umožňuje přenášet pouze malé řezné síly a při broušení dochází k uvolňování jednotlivých zrn z pojiva a tím k samoostření a zároveň opotřebování nástroje.

Při broušení dochází k plastickým deformacím a tření mezi nástrojem, obrobkem a třískami. Kvůli tomu dochází na styčné ploše k vysokým teplotám, které mohou mít za následek utavení zrna nástroje nebo tavení materiálu. Řezná rychlost brusného pásu je vysoká a může dosahovat 30-100 m/s. Doba záběru zrn je tak krátká.

1.1.1 Kinematika rovinného broušení

Podle geometrie povrchu obrobku a jeho vzájemné poloze s nástrojem se broušení dělí na několik kategorií. Zde bude jakožto jediné relevantní probráno pouze tangenciální přímočaré rovinné broušení.

Charakteristickým parametrem broušení je rychlost broušení. Ta se skládá ze vzájemné rychlosti obrobku a nástroje. Vzhledem k vysokým rychlostem nástroje je rychlost posuvu brusky vůči obrobku zanedbatelná a za řeznou rychlost se tedy považuje pouze obvodová rychlost nástroje:

$$v_c = \frac{\pi \cdot d_s \cdot n_s}{60 \cdot 1000} \quad (1.1)$$

kde d_s je průměr brousicího kotouče v mm a n_s je rychlost otáčení brusky v ot/min. Při běžném broušení je dosahováno rychlostí 30 až 35 m/s. [1][2]

Brusný pás má s obrobkem úzkou styčnou plochu, kde se ho dotýká pouze v tečném místě vodícího bubnu pásu a plechu, a to po celé šíři pásu. Styčná plocha má tedy tvar úzkého obdélníku širokého jako pás a dlouhého podle styčné plochy. Délku styčné plochy nelze určit přesně, protože na její velikost májí vliv proměnné a těžko určitelné faktory jako elastické deformace vodícího bubnu pásu podle přítlaku

brusky či elastické deformace pásu podle přítlaku nebo rychlosti otáčení. Z této skutečnosti vyplývají směrová omezení pohybu brusky, kdy je výhodnější s bruskou pohybovat pouze v tangenciálním směru otáčení pásu, kde je předem známa šířka broušené cesty.

1.2 Upravovaný materiál

Obrobkem je zde plech o rozměrech až 2 x 3 m, tloušťky až 40 mm. Plechy mohou být ze skladu deformované, a proto při položení na pracovní plochu stroje nelze spoléhat na konstantní výšku pohybu broušící hlavy. Materiál plechů je především nerezová ocel. Níže jsou vyobrazeny fotografie možných tvarů plechů, pro jejichž obroušení je stroj navrhován.

U broušených plechů se předpokládá vysoká hmotnost, a proto by se při broušení, ať už silou tahu brusky či kvůli vibracím, neměly samovolně přemísťovat. Návrh algoritmu broušení bude založen na tomto předpokladu. Změna polohy plechu při broušení by mohla vzhledem k použitému algoritmu znamenat nesprávné obroušení. Pokud by se plechy hýbaly, bude potřeba vyřešit jejich dodatečné ukotvení.



Obrázek 1.1: Ukázka broušeného materiálu 1



Obrázek 1.2: Ukázka broušeného materiálu 2
plech je broušen před stočením

2 Bruska

Pro úpravu povrchu bude použita pásová bruska na kov GRIT GX 75. Bruska byla vybrána po konzultaci se zadavatelem na základě jeho předchozích zkušeností.

Tabulka 2.1: Parametry pásové brusky

Otáčky na prázdno	[ot/min]	3000
Rozměr pásu	[mm]	75 x 2000
Průměr vodícího bubnu pásu	[cm]	20
Výkon	[kW]	3
Napětí	[V]	380
Hmotnost	[kg]	75



Obrázek 2.1: Pásová bruska GRIT GX 75

Řezná rychlost brusky je stanovena na základě vztahu (1.1):

$$v_c = 31,4 \text{ m/s}$$

Rychlost brusky daná výrobcem je považována za ideální a zároveň odpovídá běžným teoretickým rychlostem, a proto nebude použit frekvenční měnič pro řízení rychlosti brusného pásu. Místo úpravy rychlosti pásu bude pro dosažení požadovaného stupně obroušení řízena rychlost posuvu, resp. čas brusky strávený na úseku cesty broušení.

3 Detekce obrysu plechu

Při broušení je nutná obezřetnost v místech hran plechu, protože při sjetí nebo najetí bruskou přes hranu by mohla být hrana sražena, což je nežádoucí. Z tohoto důvodu je nutná detekce hran plechu. Obráběný plech může mít různou velikost a být neobdélníkového neplného obrysu - plech může mít výřezy různého tvaru a velikosti nebo vrtané díry.

Původní návrh zadavatele byl definice několika souřadnic, kde se plech nachází, na panelu, a stroj by na základě těchto přibližných souřadnic plech obrousil. Takto sice řízení realizovat lze, nicméně není zaručena kvalita obroušení ve smyslu vyhnoutí se okrajům plechu a obroušení celé plochy plechu. Detekce tedy bude vycházet z několika výchozích bodů a dále bude prováděno měření snímači na hlavě stroje pro přesnější nalezení hran. Malé díry velikosti řádově zlomku šířky brusného pásu budou ignorovány a přebroušeny, protože zde díky malé velikosti nehrozí zničení hran.

Pro snazší rozlišení plechu od podložky snímači bude vhodné podložku od plechu vhodným způsobem oddělit podle způsobu snímání. Bude-li plech detekován optickým snímačem, bude vhodné mít podložku jiné světelné odrazivosti, než bývá plech. Bude-li detekován indukčním snímačem, bude vhodná podložka například ze dřeva. Jiný, nekovový, materiál zároveň slouží jako prevence proti poškrábání spodní strany plechu.

3.1 Snímače pro detekci obrysu plechu

Pro snímání obrysu předmětu existuje několik postupů. V dnešní době rapidního rozvoje informačních technologií se jako první nabízí využití kamery a počítačového vidění. Tímto způsobem lze obrys plechu detekovat vcelku snadno aplikací některého z algoritmů rozpoznání hran. Použitelnost tohoto způsobu je však silně závislá na světelných podmínkách, kde mohou mít silně negativní vliv odlesky, protože jde o kovový materiál. Dalším problémem je zde zabránění celé pracovní plochy, kde upevnění kamery může být z hlediska dodatečných konstrukcí nepraktické. Příkladem budiž řešení, kde je kamera umístěna kolmo k pracovní ploše a tím by musela být relativně vysoko nad pracovním stolem, což vyžaduje přídavné konstrukce, nebo by kamera mohla být upevněna konstrukčně méně náročněji například na sloupku na kraji stroje a plech snímala z boku. Druhý způsob však negarantuje přesné rozlišení výřezů kvůli úhlu pohledu a parametrům kamer v relevantní cenové kategorii a je náchylnější na odlesky. Nevýhodou detekce hran s využitím zpracování obrazu

je vysoká výpočetní náročnost.

Další možností je získání obrysu za použití senzorů vzdálenosti nebo přiblížení. Zde se pohybuje senzorem nad pracovní plochou a hlídá se přítomnost materiálu v daném místě. Tato metoda zahrnuje využití manipulačního mechanismu stroje a jejím záporem je především časová náročnost, která je úměrná rychlosti polohovacímu mechanismu, senzorické výbavě a mapovacím algoritmu. Sensory přiblížení (indukční) jsou ekonomicky výhodnou variantou, avšak mají krátký dosah (v řádu centimetrů). Pro první testování zde budou použity optické (laserové) snímače vzdálenosti. Důraz zde musí být brán také na odolnost snímače vůči kovovému prachu a vibracím.

3.2 Plánování trasy

Obroušení plochy lze docílit více způsoby. Pro tento účel byly uvažovány dva způsoby. Jednodušší možnost je brousit materiál, a při tom pomocí senzorů kontrolovat přítomnost plechu před brousicí hlavou. Nachází-li se před brousicí hlavou díra, je bruska zvednuta. Broušení pokračuje v místě, kdy se pod brusku opět nachází materiál. Nedostatkem tohoto řešení je možnost méně kvalitního rozlišení hran a případné nedokonalé obroušení.

Druhý možný přístup je založen na detekci hran před začátkem broušení. Detekce hran umožňuje poskytnout jejich dokonalejší rozlišení, avšak za cenu větší senzorické výbavy. Nevýhodou tohoto způsobu je vysoká časová náročnost hledání hran před začátkem broušení. Pro vyšší kvalitu obroušení byl zvolen druhý přístup.

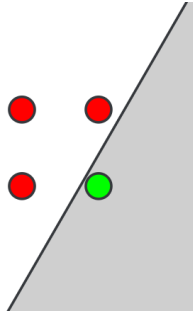
3.3 Algoritmus detekce hran

Detekce hran na ploše pomocí snímačů vzdálenosti je netriviální úlohou, pro kterou byla provedena rešerše již dostupných algoritmů. Výsledek rešerše byl však negativní, kdy se žádný dostupný algoritmus nepodařilo dohledat. Bylo proto přistoupeno k tvorbě vlastního. Požadavky na algoritmus jsou časově optimální detekce hran, výpočetní náročnost schůdná pro použití na průmyslovém řídicím systému.

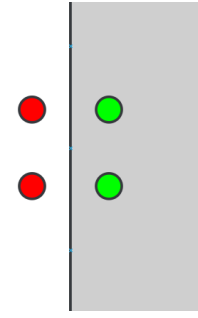
Prvním uvažovaným způsobem mapování je procestovat celou pracovní plochu a zaznamenávat přítomnost plechu. Toto "skenování" je však časově extrémně náročné. Další uvažovaný způsob hledá a sleduje hrany. Ty by však bylo obtížné a opět časově náročné detekovat pouze s jedním senzorem, a proto zde bylo přistoupeno k využití matice čtyř senzorů pro snazší rozlišení hran. Podle množství a polohy sepnutých snímačů se poté rozhoduje o přítomnosti hrany a směru dalšího pohybu. Tento algoritmus bude simulován v prostředí Matlab pro ověření jeho funkčnosti.

3.3.1 Detekce směru hrany

Algoritmus vyčítá přítomnost plechu za použití čtyř senzorů uspořádaných do čtvercové matice. Na základě kombinace sepnutých snímačů lze určit, zda je hrana plechu uprostřed matice nebo na její kraji a dále lze určit směr tečny hrany.



(a) Jeden senzor nad plechem



(b) Dva senzory na plechem

Obrázek 3.1: Detekce hrany plechu maticí čtyř snímačů, plech znázorněn šedou barvou

Obecně lze rozlišení r detekce směru určit vztahem $r = 2 \cdot n$ kde n je počet snímačů uspořádaných do pravidelného mnohoúhelníku (v případě čtyř snímačů je to čtvercová matice). Čtyři snímače tedy dokážou rozlišit osm směrů tečny hrany: $r = 2 \cdot 4 = 8$. Na základě množství sepnutých snímačů lze definovat váhu m udávající množství snímačů nad plechem:

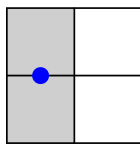
$$m = \sum_{i=1}^n s_i \quad (3.1)$$

kde s_i je stav i -tého snímače (nachází-li se pod ním plech, je stav roven 1, jinak 0). Směr tečny hrany lze odvodit na základě polohy těžiště snímačů. Těžiště $COM_{x,y}$ je vypočítáno ze stavu snímačů, váhy a jejich poloh (x_i, y_i) :

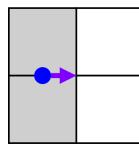
$$COM_x = \frac{\sum_{i=1}^n s_i \cdot x_i}{m} \quad (3.2)$$

Obdobně pro y . Z polohy těžiště vůči středu matice snímačů je odvozen gradient stavu snímačů. Konečně, tečna na hranu plechu je kolmice ke směru gradientu.

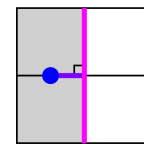
Matice senzorů je dále vyobrazena jako mřížka 2 x 2, kde přítomnost plechu pod senzorem je značena šedou barvou.



(a) Těžiště (modrá)



(b) Gradient (fialová)

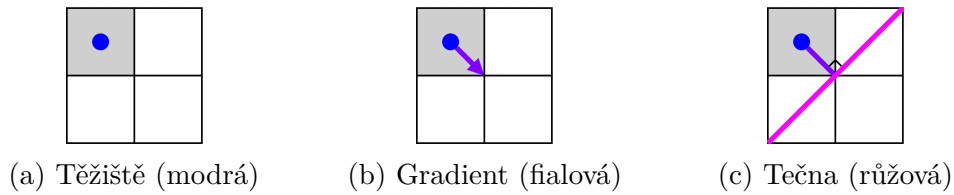


(c) Tečna (růžová)

Obrázek 3.2: Postup určení tečny hrany - 2 senzory nad plechem

3.3.2 Určení směru pohybu snímačů

Pro jednodušší rozhodování o směru budoucího pohybu byl pohyb při hledání hrany omezen pouze na jediný směr a to levotočivý (*levo nebo pravotočivý směr nepředstavují jeden proti druhému žádnou výhodu ani nevýhodu*). Snímač postupuje podél

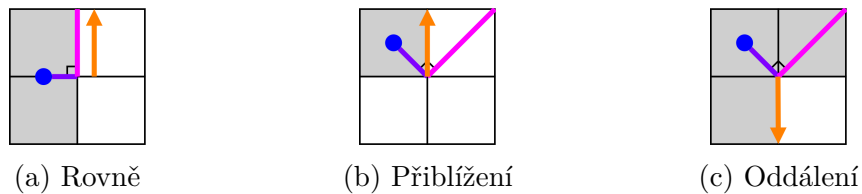


Obrázek 3.3: Postup určení tečny hrany - 1 senzor nad plechem

hrany a při tom jsou počítány výše uvedené údaje - váha m a těžiště $COM_{x,y}$ - a podle nich je rozhodováno o směru hrany. Cílem je udržet snímače na hraně plechu (jinými slovy hranu plechu uprostřed pole snímačů).

Podle směru hrany a váhy je rozhodováno o směru následného pohybu. Obecně, je-li váha rovna polovině počtu snímačů $m = n/2$, tedy plech se nachází uprostřed matice snímačů, směr následujícího pohybu je dán ve směru rozeznané tečny hrany. Je-li váha menší než polovina počtu snímačů $m < n/2$, předpokládá se, že snímače sjíždějí z plechu a směr pohybu v dalším kroku je směřován zpět na plech. Opačně, je-li váha větší než polovina počtu snímačů $m > n/2$, předpokládá se, že snímače najíždějí na plech a další krok je směřován z plechu ven.

Konkrétně na naší matici čtyř snímačů a při levotočivém pohybu je směr omezen pouze na tři možnosti podle váhy. Při váze snímačů $m = 2$ je směr hrany považován za směr pohybu rovně - pohyb je směřován doleva od směru gradientu, tedy pouze po ose X nebo ose Y. Při váze $m = 1$ není aplikován šikmý směr tečny hrany, ale je ponechána pouze složka pohybu zajišťující přiblížení blíže k plechu. Při váze $m = 3$ opět není ponechán šikmý směr ve směru tečny, ale je aplikována pouze složka pohybu zajišťující oddálení od plechu. Za pohyby přibližování/oddalování stojí snaha udržet střed snímačů na hraně plechu.



Obrázek 3.4: Postup určení směru pohybu (oranžová je směr pohybu)

Speciálním případem jsou situace, kdy je váha $m = 0$ nebo $m = 4$, což znamená, že se všechny snímače nachází nad plechem nebo mimo něj a hrana byla ztracena. Tato situace může nastat například při přejíždění děr, a proto se pro tyto případy uvažuje zachování posledního směru pohybu.

3.4 Mapování obrysu

Před začátkem broušení je třeba zmapovat obrys plechu pro určení jeho polohy, protože vstup od operátora nemusí být přesný a pokud by se bruska řídila pouze na

základě onoho vstupu, nemusí být zajištěno obroušení celé plochy, protože operátor může vstup zadat nepřesně.

3.4.1 Nalezení první hrany

Hledání začíná zadáním přibližných souřadnic hranic plechu operátorem. Z těchto hranic se vypočte přibližný střed plechu. Sensory se potom dají do pohybu z výchozí pozice k předpokládanému středu plechu. Pohyb trvá až do chvíle, než je pod senzorem rozpoznán plech.

3.4.2 Nalezení vnějšího obrysu plechu

Po nalezení hrany se senzory vydají po hraně a s využitím výše zmíněného algoritmu detekce hran se analyzuje přítomnost hrany a určuje se směr následujícího pohybu. Hledání je ukončeno v okamžiku, kdy jsou senzory blízko výchozího bodu hledání a zároveň se pohybují ve směru, kdy se k němu přibližují.

Trasa senzorů, resp. trasa kolem hrany, je považována za hranu. Seznam projitých souřadnic je kvůli snížení výpočetní náročnosti v budoucích krocích zdecimován a zjednodušen. Ze zjednodušené cesty je vytvořen uzavřený mnohoúhelník (dále jako polygon). Díky přeměně uzavřené cesty na polygon lze poté při broušení zjišťovat, zda se bruska nachází nad plechem nebo mimo, resp. využijí se metody pro zjišťování přítomnosti bodu v polygonu.

3.4.3 Hledání děr v plechu

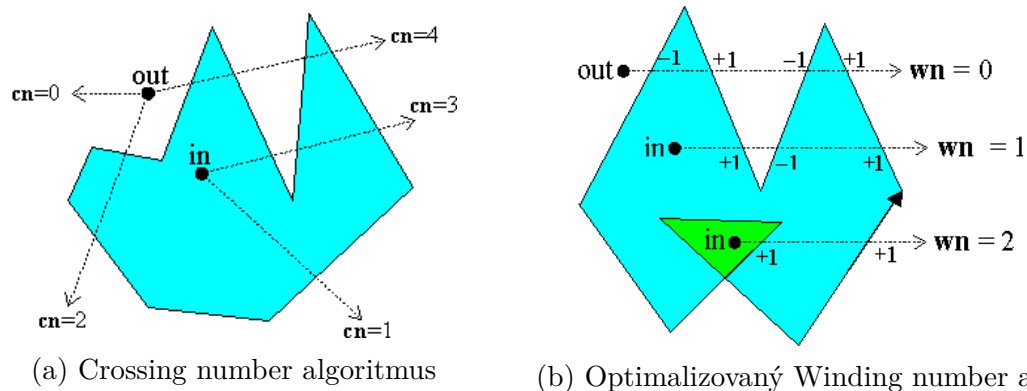
Hledání děr v plechu probíhá kvůli časové úspoře při broušení, ke kterému je přistoupeno po nalezení vnějšího obrysu plechu. To je prováděno vždy na přímkách kolmých k tangentě brusného pásu. Bruska tedy jezdí pouze dopředu a dozadu a trajektorie broušení je série paralelních cest. Matice senzorů jsou od brusky odsazeny do boku a směr pohybu broušení je koncipován tak, aby se matice senzorů pohybovala ve smyslu broušených pruhů vždy před bruskou. Každý pruh je tedy vždy nejdříve celý oskenován senzory a až poté přebroušen.

Je-li při broušení detekována hrana, o které se nepředpokládá, že je hranou vnějšího obrysu, je broušení přerušeno a je znovu použit algoritmus pro hledání hran. Tentokrát je polygon hrany uložen jako díra. Podle vlastností díry je poté rozhodnuto, zda bude ignorována a přebroušena nebo nad ní bude bruska zvednuta. Po nalezení díry se opět pokračuje v broušení.

3.4.4 Bod v polygonu

(*Point in polygon - PIP*). Pro určení, zda se bod nachází v polygonu se používají dva přístupy. Oba fungují pro konvexní i konkávní polygony. *Ray casting* algoritmus (známý také jako Crossing number) vyšle ze zkoumaného bodu paprsek (polopřímku) libovolným směrem a poté prochází všechny hrany polygonu a zkoumá, zda se protínají s paprskem. Je-li počet protnutí lichý, znamená to, že je zkoumaný bod

uvnitř polygonu, je-li počet sudý, je bod vně. *Winding number* algoritmus (známý také jako Nonzero-rule) prochází všechny body polygonu a počítá, kolikrát se kolem něho polygon obtočil. Toto vyžaduje mnoho výkonu pro počítání úhlů. Pro algo-



Obrázek 3.5: Princip algoritmů určení přítomnosti bodů v polygonu [4]

ritmus existuje optimalizace, která, stejně jako předchozí algoritmus, vyšle paprsek a nyní počítá kolik stran bylo protnuto po a proti směru hodinových ručiček. Směry se od sebe odečtou a výsledné číslo udává, kolikrát se kolem bodu polygon obtočil. Oba algoritmy jsou s optimalizacemi srovnatelně rychlé. Nonzero-rule umožňuje určit polohu bodu spolehlivěji, a proto by měl být používán spíše ten. [4]

3.5 Rozmístění snímačů

Matice snímačů by měla být odsazena od brousicího pásu do boku takovým způsobem, kdy při broušení plechu bude vždy sledovat prostor dalšího broušeného pruhu a algoritmus tak bude moci hlídat přítomnost děr v zatím neobroušené ploše.

Vzájemná vzdálenost snímačů matice udává přesnost, s jakou bude možné rozpoznávat přítomnost hran. Zde je žádoucí umístit snímače v nejmenší možné vzájemné vzdálenosti.

Velký vliv na výslednou přesnost zmapování plechu má umístění matice vzhledem k výškově polohovatelné hlavě brusky. Polohování je totiž zajištěno otáčením na kloubu a při zvedání se tak kromě zdvihu mění i poloha v ose X. Senzory by se měly nacházet blízko místa kde dochází k broušení a lze je ukotvit buďto k samotné brousicí hlavě nebo k platformě manipulátoru. Oba způsoby mají své nevýhody. Při zavěšení na brusce nastává problém, kdy se při zvedání brusky zvedají současně i senzory. Při zvedání se tedy mění poloha snímačů v souřadnici X. Zároveň jsou při zvednuté brusce senzory daleko od plechu, což snižuje rozlišení snímačů a výběr snímačů pouze na typy schopné snímat na vzdálenosti řádově až desítky centimetrů. Zároveň snímače mohou být zatíženy šumem vznikajícím při přenášení vibrací ze spuštěné brusky. Druhý způsob, ukotvení k platformě manipulátoru, má nevýhodu ve změně vzájemné polohy snímačů a brusky při zvedání. Zde se však jedná prakticky o jedinou nevýhodu, kterou lze snadno kompenzovat bez přidání další sensoriky zadáním tloušťky plechu před startem programu. Na základě výšky plechu se

vypočte odsazení snímačů a brusky v ose X. Sice se předpokládá broušení deformovaných plechů, nicméně deformace s vysokou pravděpodobností bude nízká tak, že odchylka bude v řádech milimetrů, a tedy v tomto případě zanedbatelná. Vzhledem ke znatelně menším uvažovaným nevýhodám byla zvolena druhá varianta, kdy bude matice snímačů vetknuta přímo na platformě manipulátoru.

3.6 Ověření algoritmu hledání hran a plánování trasy simulací

Pro účely vývoje a ověření funkčnosti algoritmu bylo využito prostředí Matlab. Simulace se však liší od reálného systému. Jejím omezením je pracovní plocha s plechem vyjádřená ve formě bitmapu, kdy lze vyčítat přítomnost plechu z jednotlivých pixelů. Pohybovat se lze po kvantech, nikoliv spojitě.

Následuje popis činnosti programu podpořený úryvkem pseudokódu. **Proměnné apod. se v pseudokódu kvůli zjednodušení a vysvětlení jejich účelu mohou jmenovat jinak. Hodnoty vyjádřené neurčitými slovy jako malé, blízké apod. jsou parametrické hodnoty, jež mohou být měněny během ladění programu.*

3.6.1 Pomocné funkce

V programu je často využíván výpočet váhy snímačů plechu. Pro tuto úlohu byla vytvořena funkce `getMass()`. Funkce vrací váhu snímačů podle (3.1).

```
1 | function mass = getMass ()
2 |     mass = sum(sensor_states);
3 | end
```

Zdrojový kód 3.1: Funkce `getMass()`

Další zjednodušení je realizováno funkcí `drawHead()` usnadňující vykreslování vizualizačních (debugovacích) elementů, resp. pozic brusky a senzorů. Do argumentů funkce jsou zadávány *logical* hodnoty určující, které prvky budou vykresleny - poloha brusky, obrys styčné plochy brusky, poloha senzorů, obrys matice senzorů - funkce vykresluje body a obdélníky.

3.6.2 Nalezení první hrany

Po spuštění skriptu je zobrazena bitmapa plechu. Následuje volání funkce `getBoundaries()`, která nechá uživatele klikáním do náhledu plechu zadat jeho přibližné hranice. Následně se z minim a maxim souřadnic zadaných bodů určí rámeček přibližné pozice plechu.

```
1 | function getBoundaries ()
2 |     for i=1:4
3 |         boundPoints(:, i) = ginput(1); %input points
4 |     end
```

```

5 | left = min(boundPoints(1,:));
6 | right = max(boundPoints(1,:));
7 | bottom = max(boundPoints(2,:));
8 | top = min(boundPoints(2,:));
9 | end

```

Zdrojový kód 3.2: Funkce getBoundaries()

Se znalostí přibližné pozice plechu funkce *findFirstEdge()* pohybuje brusku do středu plechu. Pokud v jedné souřadnici dorazí do poloviny plechu, pak se přibližování omezí na pohyb v jedné ose. Ve chvíli, kdy váha matice senzorů udává nalezení hrany, je hledání ukončeno.

```

1 | function findFirstEdge()
2 |     dir = move diagonally;
3 |     onEdge = false;
4 |     while(~onEdge)
5 |         if (one position axis at sheet center)
6 |             dir = move along second axis;
7 |         end
8 |         pos = pos + dir;
9 |         mass = getMass();
10 |        onEdge = (mass > 0) && (mass < 4);
11 |     end
12 | end

```

Zdrojový kód 3.3: Funkce findFirstEdge()

3.6.3 Nalezení vnějšího obrysu plechu

Funkce *findOutline()* slouží k vyhledávání obrysu hran. Výchozí bod je pozice, kde jiná část programu narazila na hranu. Funkce mění pozici hlavy a postupně objede celý obrys plechu. V každém kroku je vypočtena váha senzorů a těžiště a na jejich základě je rozhodováno o budoucím směru pohybu hlavy se senzory. Když se hlava přiblíží k počáteční pozici, je hledání ukončeno.

```

1 | function findOutline()
2 |     outline(:,1) = pos;
3 |     edgeStart = pos;
4 |     foundEdgeStart = false;
5 |     step = 2;
6 |     foundOutline = false;
7 |     while(~foundOutline)
8 |         presence = sensors matrix;
9 |         mass = getMass();
10 |        rotate presence based on its mass to influence direction;
11 |        mass = sum(presence);
12 |        com = calculate from presence; %center of mass
13 |        dir = calculate from com;
14 |

```

```

15     pos = pos + dir;
16     outline(:, step) = pos;
17
18     if (position is getting near to edgeStart && position is
19         near enough to edgeStart)
20         foundOutline = true;
21     end
22     step = step + 1;
23 end

```

Zdrojový kód 3.4: Funkce findOutline()

3.6.4 Zjednodušení hrany

Funkce *simplifyOutline()* byla vytvořena výhradně pro snížení výpočetních nároků při používání funkce pro zjišťování přítomnosti bodu v polygonu. Později zde bylo implementováno i filtrování děr. Funkce je používána pro zjednodušení jak vnějšího obrysu, tak děr. Na začátku je sada bodů obrysu převedena na polygon. Pokud je polygon malý, je set bodů ponechán, jinak je set bodů decimován. Dále jsou vyfiltrovány tenké části polygonu (tenké a malé části mohou být přebroušeny, protože zde nehrozí, že bruska zničí jejich okraje). Dále je zjednodušený set bodů převeden na polygon, čímž je dále zjednodušen (vynechání bodů ležících v jedné přímce). Pokud je zjednodušován vnější obrys, je k němu vypočtena konvexní skořepina (lze si představit jako omotání tvaru provázkem), která je při broušení využívána jako hranice plechu. Pokud je zjednodušována díra, je na základě její velikosti rozhodováno, zda je malá a bude pro broušení ignorována či nikoliv.

```

1  function simplifyOutline()
2      outpgon = polyshape(outline);
3      if outpgon.area is not small
4          simplifiedOutline = decimate set of outline points;
5      else
6          simplifiedOutline = outline;
7      end
8      pgon = polyshape(simplifiedOutline);
9      filter parts of outline that or too small or thin; %polybuffer
10     simplifiedOutline = pgon.Vertices;
11     if (simplifying outline)
12         outlinePolygon = pgon;
13         convexHull = convhull(pgon);
14     else
15         add hole to outlinePolygon;
16         decide whether hole is too small to be ignored while
17             grinding;
18     end
19 end

```

Zdrojový kód 3.5: Funkce simplifyOutline()

3.6.5 Broušení a hledání děr

Funkce *grind()* slouží pro kontrolu pohybu cyklu broušení. Při broušení jsou zároveň vyhledávány díry v plechu. Cyklus začíná v rohu hranice plechu. Bruska brousí nahoru a dolů (po ose manipulátoru). Na začátku každého cyklu, je ověřeno, zda se hlava nachází na horní nebo spodní hranici. Pokud je na hranici, pak ustane pohyb nahoru/dolů a hlava je přesouvána do dalšího broušeného pruhu. Následuje ověřování polohy brusky vůči plechu. Pokud je bruska na plechu, pak je zapnuto broušení. Zjišťování se provádí pomocí Matlab implementace algoritmu **PIP** - *inpolygon()*.

Pro brusku jsou definovány dva kontrolní body, pro které se ověřuje, zda se nachází na plechu a zároveň mimo již nalezené díry. Kontrolní body jsou umístěny přibližně ve třetinách šířky broušicího pásu, a proto i když jeden bod bude ve vzduchu, ale druhý bude stále na plechu, tak je možné brousit. Broušení je tedy zapnuto, pokud se alespoň jeden z bodů nalézá na plechu.

Při broušení se také detekují díry v následujícím broušeném pruhu. Pokud je nalezena hrana (váha senzorů je menší než čtyři), určuje se, zda se senzory nachází blízko hrany obrysu plechu nebo blízko již objevené díry. Pokud tomu tak není, je spuštěna funkce pro hledání hran *findOutline()* a následně *simplifyOutline()*. Při nalezení hrany je tedy pozastaveno broušení, nalezena hrana a poté se v broušení pokračuje.

```
1 function grind()
2     pos = bottom left corner of sheet boundary;
3     while(pos < rightBound)
4         decide whether is inside sheet or on the edge;
5         if is on up or down edge strafe to next line;
6         if strafed , go up or down;
7         pos = pos + dir;
8
9         %if on sheet – grind
10        onSheet = inpolygon(grinderPoints , simplifiedOutline);
11        if onSheet && there are already found holes
12            for %loop thru holes
13                inHole = inpolygon(grinderPoints , hole);
14                if inHole
15                    break;
16                end
17            end
18            onSheet = onSheet && ~inHole;
19        end
20        if (onSheet && moving up or down)
21            grinding = true;
22        else
23            grinding = false;
24        end
25
26        %detect holes
27        mass = getMass();
28        if mass ~= 4 %is sensor on edge
29            isInSheet = inpolygon(sensors , simplifiedOutline);
```

```

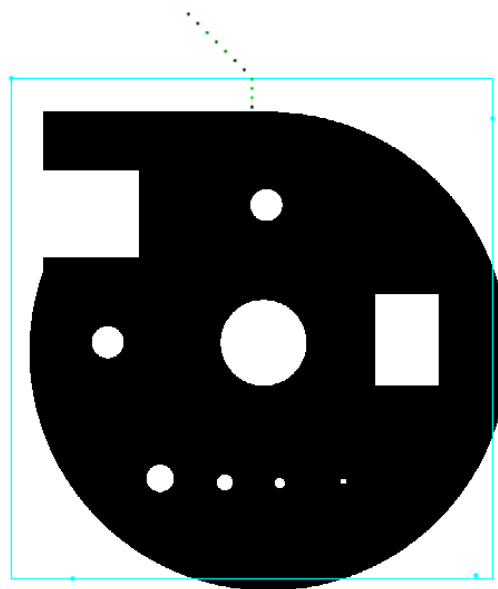
30         if (holes exist)
31             for %loop thru holes
32                 isInHole = inpolygon(sensors , simplifiedOutline);
33             end
34         end
35         if isInSheet && ~isInHole
36             findOutline ();
37             simplifyOutline ()
38         end
39     end
40 end
41 end

```

Zdrojový kód 3.6: Funkce grind()

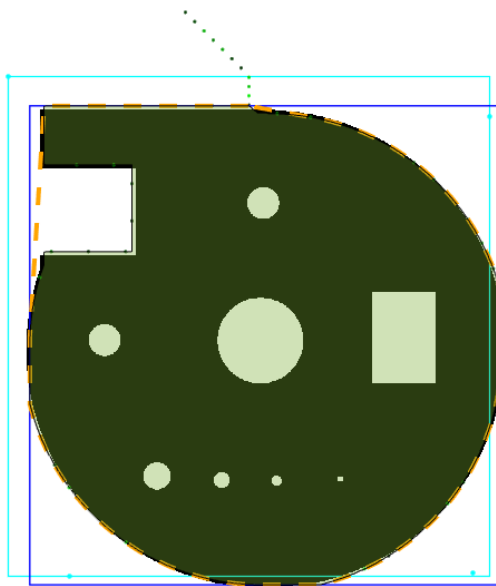
3.6.6 Průběh simulace

Workflow programu je víceméně stejná, v jakém pořadí byly uvedeny funkce. Na začátku jsou zadány přibližné souřadnice plechu pomocí *getBoundaries()*, následně je hledána první hrana *findFirstEdge()*. Po nalezení první hrany je hledán vnější obrys plechu *findOutline()* a po každém volání této funkce následuje *simplifyOutline()* zjednodušující poslední nalezenou cestu hrany. Po nalezení obrysu jsou přepočítány hranice plechu. Nakonec je volána funkce *grind()*, která zajišťuje spínání broušení podle polohy brusky vůči plechu a zároveň s odsazenými senzory hledá díry s využitím funkcí pro hledání obrysu. Mezikroky jsou zobrazeny na obrázcích.



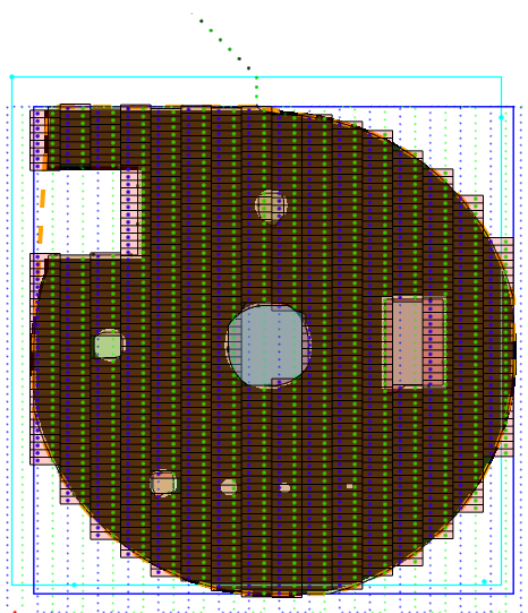
Obrázek 3.6: Zadání přibližných souřadnic a nalezení hrany

Zadané přibližné hranice plechu - tyrkysový rámeček, hledání hrany - zelené body



Obrázek 3.7: Nalezení obrysu

Cesta hledání hrany - zelené body kolem hrany, přepočítané hranice - modrý rámeček,
konvexní skořepina - oranžová čárkovaná čára, polygon obrysu - zelená plocha



Obrázek 3.8: Broušení

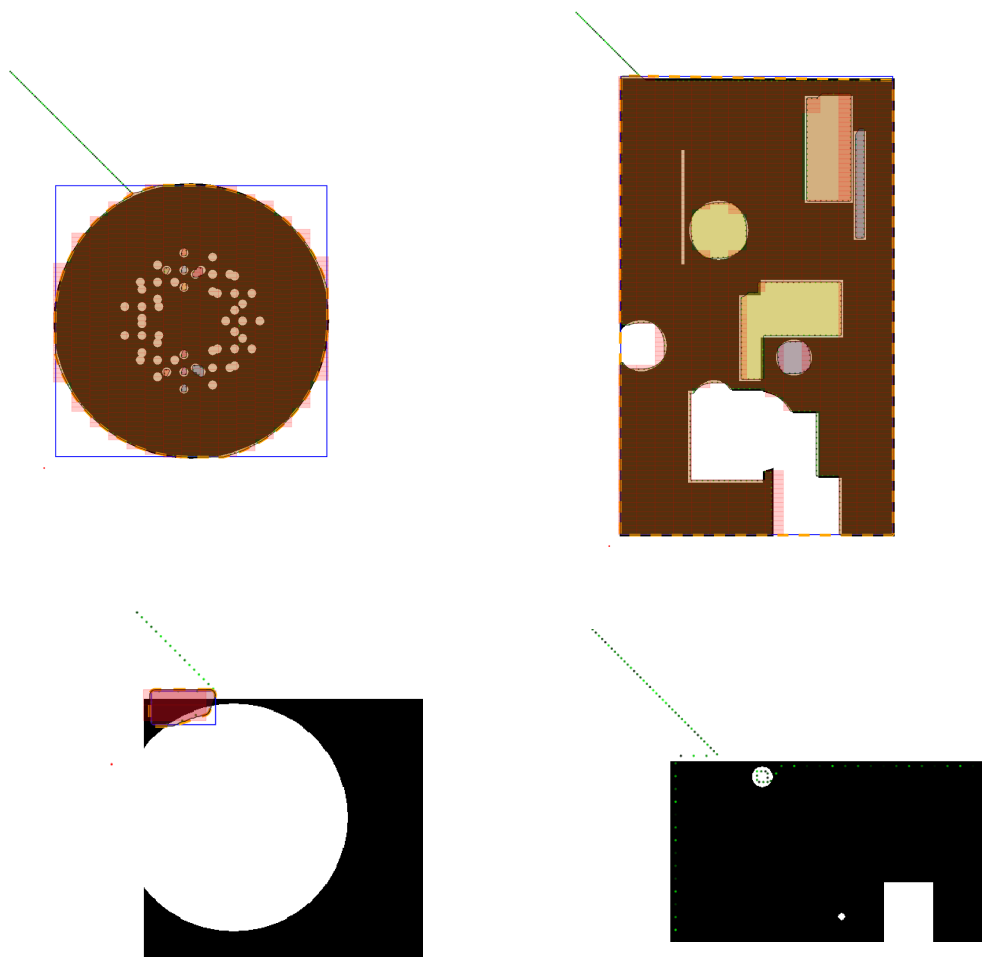
Broušení - červená plocha, cesta kontrolního bodu brusky - páry modrého a zeleného
bodu, díry - barevné plochy

3.6.7 Ověření funkčnosti algoritmu hromadnou simulací

Vývoj a simulace algoritmu probíhaly na omezeném setu testovacích ručně vytvořených tvarů plechu. Protože malý set vzorků pravděpodobně neobsahuje všechny možné tvary a situace, které mohou při provozu nastat, byl v simulaci vytvořen generátor náhodných plechů. Generátor vytvoří testovací tvar plechu obdélníkového nebo kruhového tvaru a náhodně vloží díry. Generátorem, resp. funkcí *generateSheet()*, je v simulaci nahrazeno načítání původního testovacího souboru a celá simulace je vložena do smyčky a vícekrát opakována.

Opakovaným generováním plechů s náhodným obrysem a rozložením děr je zajištěno pokrytí množství speciálních případů, které do té doby nebyly zkoušeny či uvažovány. Ačkoliv ne všechny náhodně generované plechy odpovídají potenciálním skutečným plechům, lze se přesvědčit o funkčnosti či nefunkčnosti algoritmu v mnoha případech.

Hromadným testováním byly odhaleny nedostatky algoritmu při detekci malých či úzkých děr, osamocených i ve skupinách. Na základě tohoto poznatku byla modifikována část algoritmu pro rozpoznávání děr. Při ověřování, zda je díra v plechu byl přidán postup, kdy při okamžiku nalezení hrany není okamžitě odstartováno hledání obrysu díry, ale snímače ještě krátký úsek za hranou ověřují, zda se jedná o díru dostatečně velkou, aby bylo účelné začít hledání jejího obrysu. Tímto jsou vyfiltrovány malé díry o velikosti v řádu milimetrů. Dále byla upravena podmínka, která určuje, zda vůbec bude ověřování díry provedeno. Ověřování díry je vynecháno, pokud snímače přejíždějí pouze po její hraně. Dalším poznatkem z hromadného testování bylo selhání algoritmu na úzkém rozhraní dvou děr. Toto omezení je dáno metodou detekce plechu s využitím matice snímačů. Podle popisu plechů od zadavatele většina skutečných plechů neobsahuje situace, na kterých algoritmus selhává, a proto tyto nedostatky představují pouze minoritní problém.



Obrázek 3.9: Vzorek výsledků hromadného testování

Ukázka generovaných plechů a kvality simulované detekce obrysu a broušení: Algoritmus úmyslně vynechá malé díry, které mohou být přebroušeny a nedělají mu problém různé tvary. Selhání algoritmu nastává u tenkých částí, kde je vypočten nesprávný směr dalšího pohybu a algoritmus se může zacyklit v díře.

4 Mechanické řešení stroje

4.1 Analýza mechanických nároků na stroj

Proces broušení generuje vysoké množství kovového prachu. Pokud se tento prach usazuje na funkčních mechanických členech, může způsobit jejich nadměrné opotřebení. Konstrukce stroje by tedy měla být přizpůsobena tomuto specifiku.

Rychlost pohybu os by měla být přizpůsobena pro dosažení optimálního brusného výkonu. Po konzultaci se zadavatelem byla rychlost pohybu manipulátoru brusky předběžně stanovena na 10 cm/s.

Broušení probíhá ve dvou osách, přičemž se může měnit výška. Pro osy X a Y by byl vhodný kartézský manipulátor - přímočaré posuvné na sebe kolmé členy. Jeho výhoda oproti manipulátorům zahrnujícím rotační členy je jednoduchá konstrukce a prostorová efektivita. Z důvodu nenáročnosti upevnění pramení předpoklad, že by kartézský manipulátor měl ležet na pracovním stole (tzn. bruska nebude zavěšena shora). Zvedání brusky lze realizovat třetí svislou osou nebo přes kloubový mechanismus pneumatickým válcem. Třetí osa by vyžadovala další motor a vysokou tuhost. Kvůli mechanické jednoduchosti zde byl vybrán kloubový mechanismus - bruska bude stát na kloubu a pneumatický píst bude přes páku měnit její zdvih. Nároky na stroj byly shrnuty do tabulky.

Na pracovní plochu by se měly vejít plechy o rozměrech 2 x 3 m a tloušťky až 40 mm.

Tabulka 4.1: Mechanické nároky na stroj

Konstrukce osy XY	Kartézský manipulátor
Zvedání brusky	Pneumatický válec
Rychlost broušení [cm/s]	10
Odolnost proti prachu z broušení nerez	

4.2 Rešerše existujících řešení

4.2.1 ZTPolish ZTD

Stroj pro broušení a leštění čínského výrobce určený především k odstranění vrstvy oxidu. Výrobce uvádí šířku plechu až 1500 mm a tloušťku 0,5 až 100 mm. Rychlost

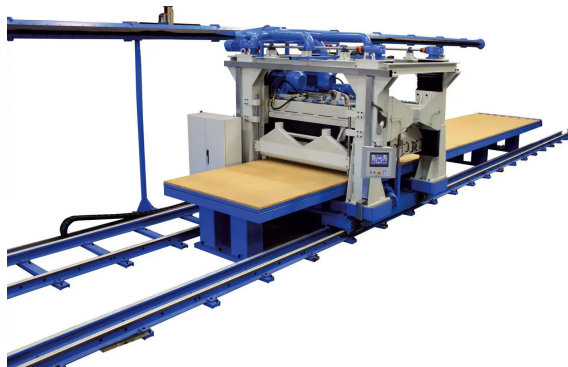
posuvu 0,5 až 2 m/min. Stroj má dvě broušící hlavy, je nabízen v několika velikostech a k broušení využívá oběhový pás. Hlavy se jeví na sobě nezávislé. Broušený materiál leží na zemi. [6]



Obrázek 4.1: ZTPolish ZTD

4.2.2 AUTOPULIT CDA-2000x12000/2UP-CP

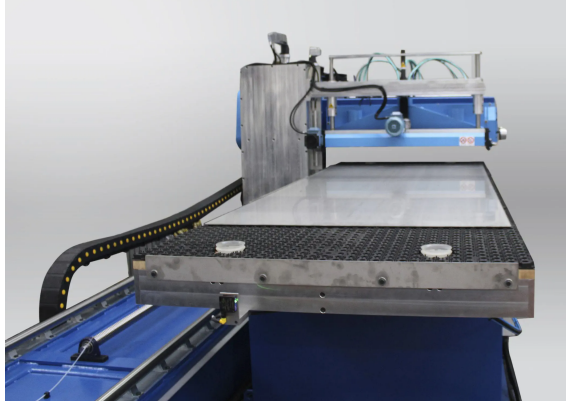
Stroj je nabízen s pracovní plochou šířky 2 m a délky 12 m. Dvě broušící hlavy jsou určeny především k leštění. Broušení neuvedeným způsobem se provádí na dlouhém stole s dřevěným povrchem zřejmě pro předejití poškrábání vyleštěného plechu. [7]



Obrázek 4.2: AUTOPULIT CDA-2000x12000/2UP-CP

4.2.3 C.O.S.M.A.P. str1 - Surface grinding machine BM

Broušení prováděno širokým brusným kartáčem přes celou šíři stolu. Tím pádem zde odpadá kontrola jedné osy. Stroj zároveň disponuje přísavkami ve stole pro znehybnění plechu při broušení. [8]



Obrázek 4.3: C.O.S.M.A.P. strl - Surface grinding machine BM

4.2.4 Shrnutí rešerše

Většina strojů pro broušení/leštění plechu využívá buď pásové brusky nebo širokého kartáče přes celou šíři stolu. Stroje jsou dělané na broušení celistvých plechů. Vedle toho existují také brousící CNC stroje umožňující broušení složitých objektů ve více osách. Zde je nicméně potřeba znát přesnou geometrii broušeného tvaru. Při rešerši jsem nenarazil na stroj umožňující broušení plechů volného tvaru a jeho automatickou detekci.

Stroje většinou používají odhalené vedení os a jeho ochrana před prachem je zajištěna odsáváním, vzájemnou polohou a vzdáleností vůči pracovní ploše. Brousící stroje jsou vesměs konstrukce portálového manipulátoru, kdy se nástroj pohybuje nad obrobkem.

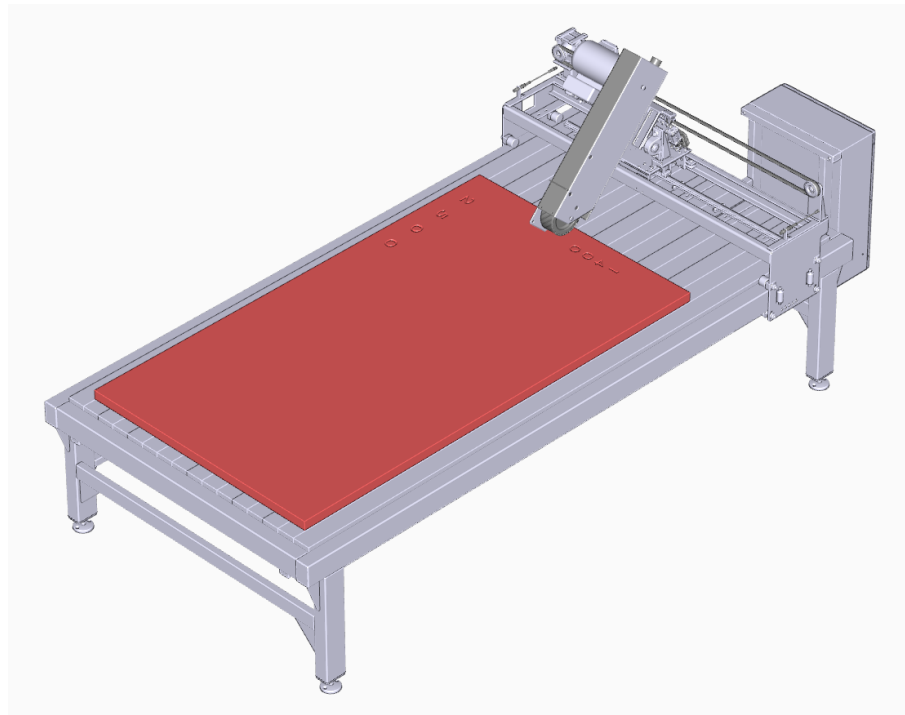
4.3 Původní studie zadavatele

V raných fázích návrhu bylo rozhodováno, zda stroj postavit podle konstrukce původní studie zadavatele či využít jiné vhodnější řešení.

Původní návrh pohonu os využívá asynchronní motory s FM a valivé vedení přímo na rámu stroje. Vedení je tedy kompletně odhalené, avšak konstrukčně jednoduché. Rozvod pohybu na osy je zajišťován řemeny. Povaha zařízení produkujícího velké množství kovového prachu však dělá tento návrh nevhodným, protože by vlivem prachu docházelo k vysokému opotřebením vodicích částí os.

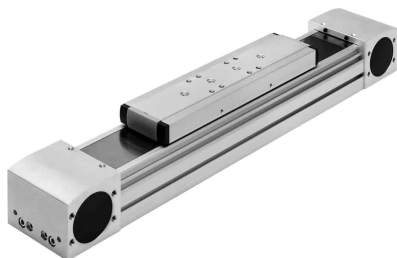
4.4 Řešení postavené na lineárním vedení Festo ELGA

Pro tuto aplikaci se nabízí jako vhodný systém vedení ELGA od Festo. Jedná se o hliníkový profil tvaru "U", kde vedení i řemeny jsou umístěny uvnitř profilu a z vrchu je profil zapouzdřen krycím pásem z pružného plechu. Celá mechanika je tak zapouzdřena. Vozík je na lineárním kuličkovém vedení. Pohony ELGA dále umožňují přivedení tlakového vzduchu do vnitřních prostor vedení a tím jeho přetlakování

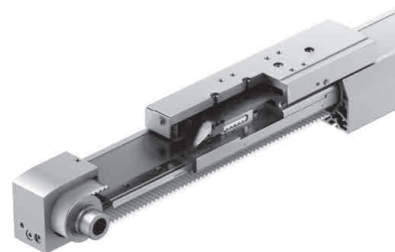


Obrázek 4.4: Původní návrh konstrukce brusky

způsobující horší proniknutí prachu skrze netěsnosti a tím vyšší životnost mechaniky vedení. Vnitřní přetlakování vzduchem je výhodné kvůli blízkosti os a brousící hlavy. Pro realizaci byl zvolen tento systém od Festo.



(a) Festo ELGA



(b) Vnitřní uspořádání ELGA-TB-KF - řemen, těsnící planžeta

Obrázek 4.5: Lineární pohon Festo ELGA

5 Pohony

Na základě analýzy stroje v předchozí kapitole byl pro pohon manipulátoru vybrán systém Festo. Manipulátor se skládá ze dvou na sebe kolmých vodorovných os: osa X, která zajišťuje pohyb po delší straně obráběcí plochy a je připevněná z obou stran obráběné plochy na rámu stroje a osa Y, na které je připevněna samotná bruska, pohybuje se po kratší straně obráběcí plochy a je připevněna na vozících osy X. Osa Y má také zdvojené vedení umožňující robustnější připevnění brusky. Touto konfigurací vzniká portálový manipulátor. Dimenzování vedení a pohonů probíhalo ve spolupráci se společností Festo a za pomoci jejich dimenzovacího software Positioning Drives. Na základě mechanických požadavků na stroj a vlastností vybrané brusky byla vytypována vedení řady ELGA, servomotory řady EMMC a servoměniče řady CMMT.



Obrázek 5.1: Návrh Festo portálového manipulátoru

5.1 Dimenzování pohonu osy X

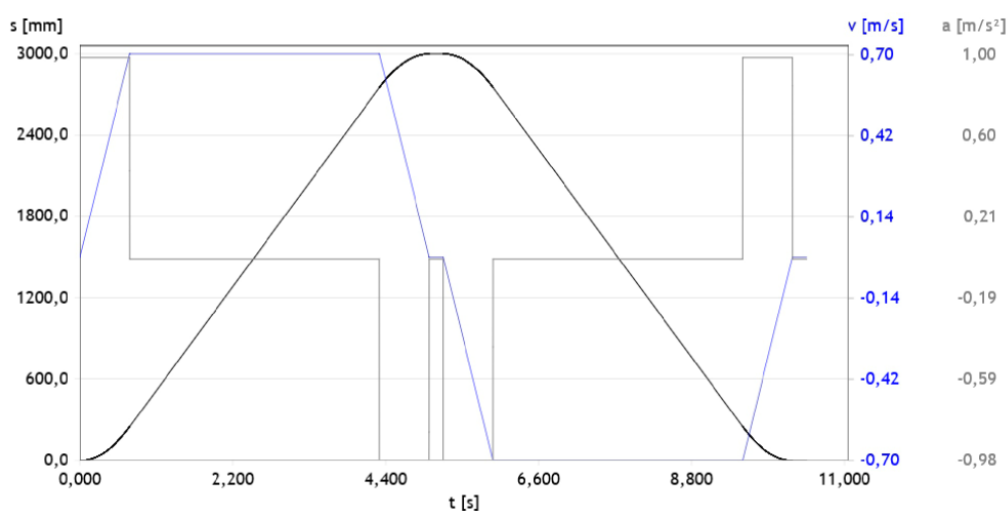
Osa X nese osu Y a brusku. Vybrány byly tyto komponenty: vedení ELGA-TB-KF-150-3000-0H spolu s axiální sadou EAMM-A-L95-80G-G2, AC servomotor EMMT-AS-80-MR-LS-RM s bezejmennou převodovkou 15:1, 110 Nm doporučenou Festo a servoměnič CMMT-AS-C4-3A-EC-S1.

Tabulka 5.1: Souhrn požadavků na osu X

Váha brusky	[kg]	75
Váha osy Y	[kg]	15
Řezná síla brusky ve směru osy X	[N]	200
Rychlost posuvu	[m/s]	0,7
Pracovní zdvih	[mm]	3000

Tabulka 5.2: Parametry zvoleného pohonu osy X

Pracovní zdvih	[mm]	3000
Rychlost	[m/s]	0,7
Jmenovité otáčky motoru	[ot/min]	3000
Jmenovitý moment motoru	[Nm]	2,2
Napětí servoměniče	[V]	230
Jmenovitý proud servoměniče	[A]	5,6
Napětí logické části servoměniče	[V]	24
Proud logické části servoměniče	[A]	0,5



Obrázek 5.2: Výsledný profil pohybu osy X

5.2 Dimenzování pohonu osy Y

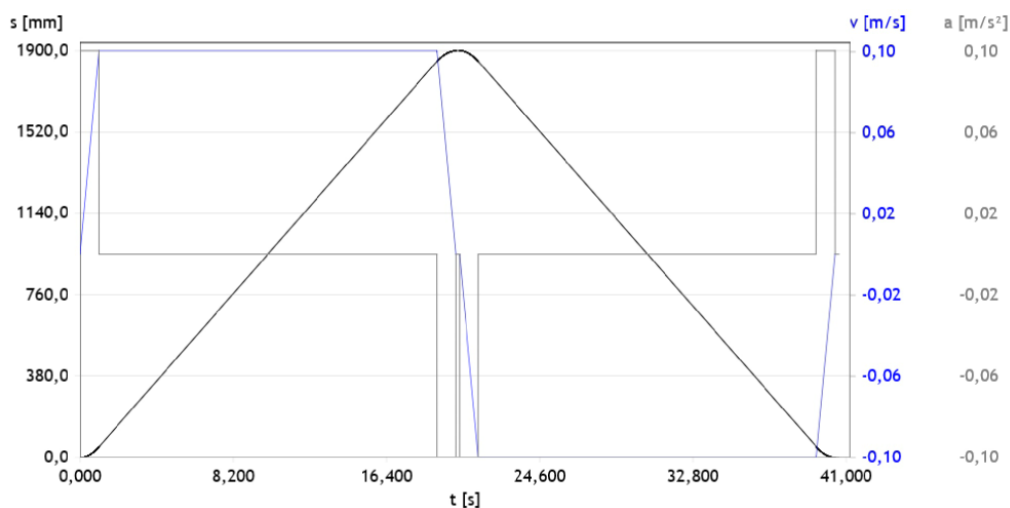
Osa Y nese pouze brusku a byly pro ni vybrány tyto komponenty: vedení ELGA-TB-KF-120-1700-0H spolu s axiální sadou EAMM-A-N80-80G, AC servomotor EMMT-AS-80-M-LS-RM s převodovkou EMGA-80-P-G5-EAS-80 a servoměnič CMMT-AS-C4-3A-EC-S1.

Tabulka 5.3: Souhrn požadavků na osu Y

Váha brusky	[kg]	75
Řezná síla brusky ve směru osy Y	[N]	0
Rychlost posuvu	[m/s]	0,1
Pracovní zdvih	[mm]	2000

Tabulka 5.4: Parametry zvoleného pohonu osy Y

Pracovní zdvih	[mm]	1700
Rychlost	[m/s]	0,1
Jmenovité otáčky motoru	[ot/min]	3000
Jmenovitý moment motoru	[Nm]	2,2
Napětí servoměniče	[V]	230
Jmenovitý proud servoměniče	[A]	5,6
Napětí logické části servoměniče	[V]	24
Proud logické části servoměniče	[A]	0,5



Obrázek 5.3: Výsledný profil pohybu osy Y

6 Elektronická část

6.1 Řídicí systém

6.1.1 Požadavky

Požadavky zadavatele projektu zahrnují robustní stroj pro povrchové opracování plochých kovových materiálů a snadnou obsluhu stroje. Na základě těchto požadavků a povahy úlohy byly určeny následující požadavky na řídicí systém:

- Ovládání skrze obrazovku s dotykovým panelem
- Pohyby z bodu do bodu (P2P)
- Dostatečný výkon pro aproximaci vzdáleností a veličin souvisejících s brusnými podmínkami
- Řešení bezpečnosti stroje

6.1.2 Porovnání řídicích systémů

Pro porovnání byly vybráni tři zástupci výrobců průmyslových řídicích systémů. Beckhoff, výrobce výkonných průmyslových PC (IPC), Siemens, výrobce nejrozšířenější řídicích systémů v ČR a B&R, výrobce moderních PLC a dle subjektivního pohledu hojně využívaný na TUL. V tabulce je uvedeno stručné porovnání těchto systémů. Kvůli převaze v technologické i praktické vyspělosti byl zvolen řídicí systém od Beckhoff.

6.1.3 Zvolený řídicí systém

Beckhoff byl vybrán jako přednostní systém používaný společností Cermitech spol. s r.o. Tento projekt slouží zároveň i jako studie využitelnosti nové vizualizace Beckhoff TwinCAT TF2000 založené kompletně na webových technologiích a využitelnosti objektové nadstavby (Object Oriented Industrial Programming - OOIP) prostředí CodeSys.

Beckhoff nabízí širokou řadu IPC běžících na různých verzích Windows pro průmyslová nebo embedded zařízení (příkladem budiž Windows 7 CE na zařízeních nejnižší výkonové třídy, nebo běžnější Windows 10 IoT). Výhodou řídicího systému

Tabulka 6.1: Porovnání řídicích systémů

	Beckhoff	B&R	Siemens
Licencování	Licence za použité funkce v cílovém HW	Roční licence na IDE, USB klíč	Licence na TIA Portal a některé funkce
Zdrojový kód	Zdrojový kód v XML, JSON	Uzavřené binární soubory	Uzavřené binární soubory
IDE	CodeSys integrovaný do Visual Studia (zdarma), díky otevřenosti možnost VSCode a další	Automation studio	TIA portal (nestabilní, chaotické)
Platforma	IPC	PLC, IPC	PLC
VCS	Git	Git	Git (omezeně, neprakticky)

běžícího na Windows je možnost snadné implementace pokročilých funkcionalit, instalace některých běžně používaných softwarů nebo i nastavení zařízení bez nutnosti připojení z externího počítače a případně i spuštění IDE přímo na zařízení.

Výraznou předností systémů od Beckhoff je také jejich cenová dostupnost a vývojové prostředí TwinCAT XAE Shell postavené na Microsoft Visual Studiu. Vzhledem k otevřenosti zdrojových souborů projektů (soubory projektu jsou ukládány jako u klasického programování ve formě textových souborů, většinou ve formátu XML a JSON, namísto ve formě binárních souborů) lze při vývoji využít i klasického verzovacího systému (Git) nebo jiných editorů (v mém případě VSCode) pro některé specifické úkony.

Model zvolený pro tuto aplikaci je Beckhoff CP6700 - IPC integrované do 10" dotykové obrazovky (tzv. panelové IPC) postavené na procesoru Intel Atom.

6.2 Vstupy a výstupy

Na zřízení se nalézá množství binárních senzorů a akčních členů. Níže se nalézá jejich výčet s uvedeným typem v kontextu připojení k PLC. Senzorika je snímána vstupními kartami s výjimkou koncových spínačů os pohonů, které jsou připojeny do servoměničů. Vzhledem k experimentální povaze stroje zatím není jisté, zda finální snímače plechu budou analogové či digitální. Prozatímní řešení je však digitální a finální konfigurace bude založena na výsledku testování. Akční členy spínané 24 V jsou připojeny na výstupní karty.

Tabulka 6.2: Souhrn vstupů

DI Koncové snímače osy X
DI Koncové snímače osy Y
DI Koncové snímače zvedání brusky
DI/AI Snímače pro detekci plechu
DI Poruchy jištění a ochran
DI Sepnutí stykačů
SI Tlačítka totalstop
SI Totalstop lanko

Tabulka 6.3: Souhrn výstupů

DO Spínání měničů
DO Spínání brusky
DO Spínání odsávání
DO Maják
DO Zvedání brusky

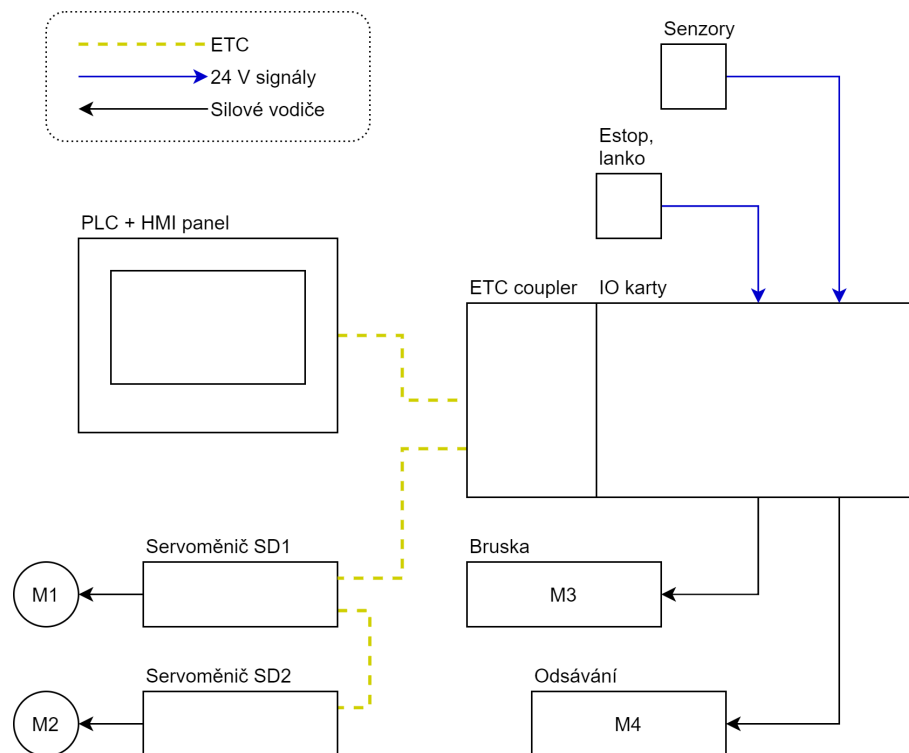
6.3 Logická část

6.3.1 Konfigurace logické části

IPC v obrazovce je propojené přes sběrnici EtherCAT s ostrovem karet vstupů a výstupů. Ostrov se skládá na prvním místě z EtherCAT coupleru, který zajišťuje připojení karet na EtherCAT a také je do něj přivedeno napájení 24 VDC ostrovu. Na coupleru jsou připojeny vstupy - karta bezpečnostních vstupů (zde jsou připojeny Total stop tlačítka a Total stop lanko) a dvě karty klasických digitálních vstupů (zde jsou připojeny pomocné kontakty jisticích prvků obvodu pro signalizaci poruchy do řídicího systému a koncové spínače pneumatického válce zvedání a snímače pro snímání hrany). Za vstupy jsou připojeny výstupy - karta bezpečnostních výstupů (zde je připojeno bezpečnostní odpínání napájení brusky a odsávání a dále 24 VDC větve bezpečného odpojení servoměničů - STO) a dvě karty klasických digitálních výstupů (připojení majáku, spínání brusky a odsávání a zapínání servoměničů - Enable vstupy). Koncové spínače servopohonů jsou připojeny do servoměničů (vstupy LM0, LM1)

6.3.2 Bezpečnostní obvody

Bezpečnostní vypnutí stroje je zajištěno jedním bezpečnostním hříbovým vypínačem a bezpečnostním tahacím lankem nataženým kolem celého stroje. Díky přítomnosti bezpečnostního lanka kolem celého stroje nejsou potřeba další nouzová tlačítka. Bezpečnostní spínače mají zdvojené kontakty připojené do dvou kanálů bezpečnostních karet. STO vstupy servoměničů jsou také dvoukanalové. Bezpečné vypnutí brusky a odsávání je zajištěno dvěma předřazenými stykači s nuceným odpínáním v sérii.



Obrázek 6.1: Přehled uspořádání hlavních elektronických komponent

6.3.3 Dimenzování zdroje 24V

Zdroj je dimenzován pro napájení 24 VDC zařízení v obvodu. Spotřeba 24 VDC

Tabulka 6.4: Spotřeba zařízení 24 VDC

Zařízení	Počet	Spotřeba [W]
IPC+HMI CP6700	1	19
Servoměnič CMMT-AS-C4-3A-EC-S1	2	12
ETC coupler EK1100	1	48
Zařízení na IO kartách	1	48

Spotřeba v proudu byla přepočtena na výkon

prvků je suma jejich spotřeby

$$P_{Sp} = \sum_{prvky} P_i \quad (6.1)$$

$$P_{Sp} = 139 \text{ W}$$

Výkon zdroje je určen podle spotřeby zařízení a předimenzován o 50 %

$$P_{DC} = P_{Sp} \cdot 1,5 \quad (6.2)$$

$$P_{DC} = 208,5 \text{ W}$$

Na základě výpočtu byl zvolen zdroj nejbližšího vyššího výkonu - 240 W, 10 A. SS obvod je jištěný a větvený elektronickými pojistkami 24 V nastavitelnými do 6 A.

6.4 Silová část

6.4.1 Bruska a odsávání

Motor brusky je třífázový o výkonu 3 kW. Z toho vyplývá proud na jedné fázi 5,6 A. Pro spínání byl zvolen stykač 10 A a motorový chránič 6-10 A. Motor odsávání je třífázový 2 kW. Pro spínání byl zvolen stykač 6 A a motorový chránič 5 A. Motory jsou napájeny z větve jištěné jističem typu D13A a dvěma v sérii zapojenými bezpečnostními stykači 18 A.

6.4.2 Pohony os

Servoměniče obou os jsou stejného typu. Silové obvody jsou napájeny jednofázově 230 V, $I_n = 4$ A. Oběma měničům jsou předřazeny jističe typu C10A. Motory pohonů jsou s měniči připojeny speciálním kabelem od Festo.

6.4.3 Dimenzování přívodu

Přívodní vodiče jsou dimenzovány pro napájení zařízení připojených na fázové napětí. Jednofázová zařízení byla rozdělena mezi fáze tak, aby byly fáze zatěžovány

Tabulka 6.5: Spotřeba zařízení napájených síťovým napětím

Zařízení	Připojení	Počet	Spotřeba [A]
Zdroj 24 VDC	1f	1	1,3
Servoměnič CMMT-AS-C4-3A-EC-S1	1f	2	5,6
Bruska	3f	1	5,77
Odsávání	3f	1	3,74
Servisní zásuvka	1f	2	3

Spotřeba ve wattech byla přepočtena na ampéry

rovnoměrně. Zátěžový proud byl vypočten jako součet proudů 3f zařízení a součet proudů nejzatíženější fáze

$$I_N = \sum_{3f} I_{Ni} + \sum_{1f} I_{Nj} \quad (6.3)$$

$$I_N = 18,11 \text{ A}$$

Síla přívodního kabelu a fázových vodičů byla podle normy ČSN EN 60204-1 zvolena 4 mm². Síla vodičů od jištění ke spotřebiči byla volena podle proudové zátěže konkrétního zařízení.

7 Pneumatická část

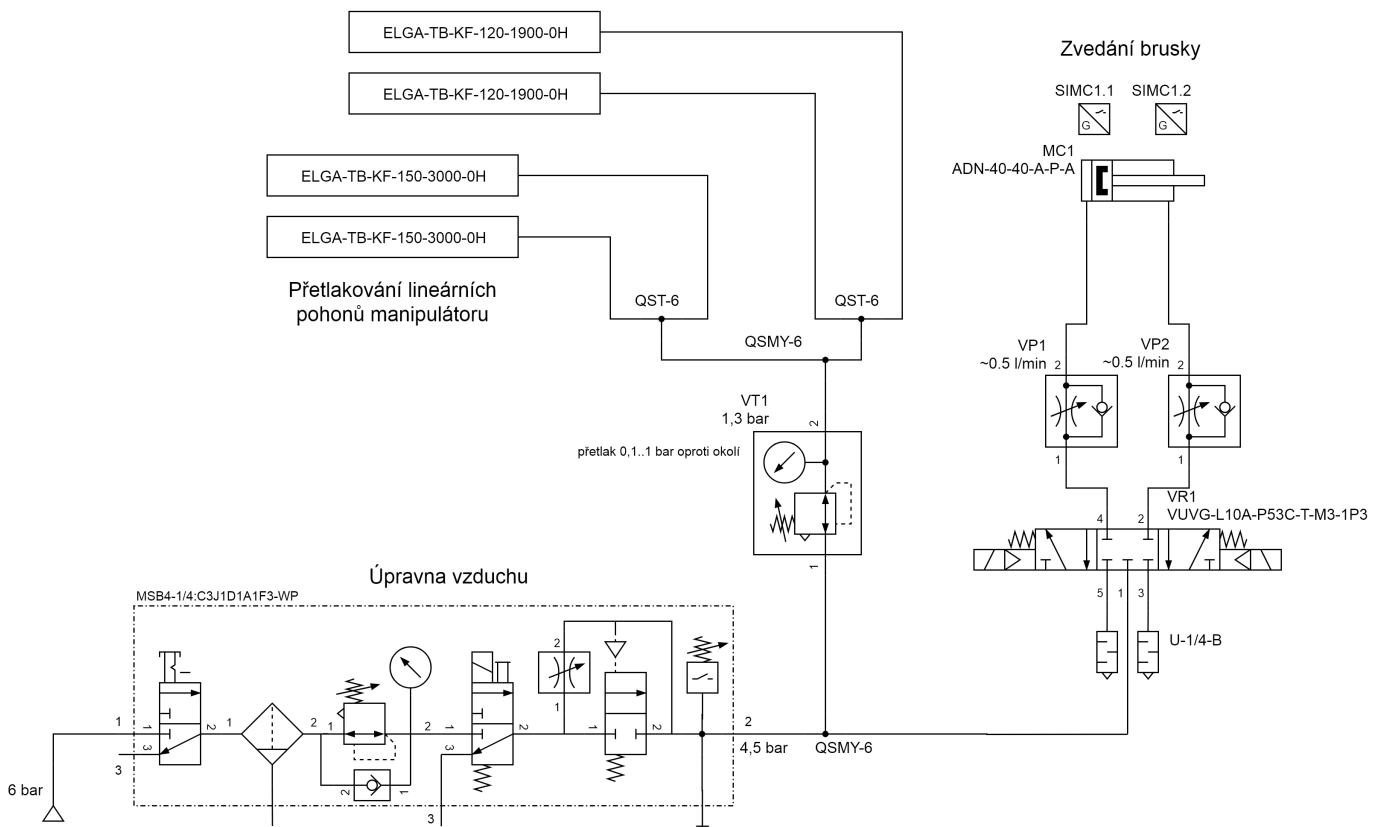
Schéma pneumatického obvodu bylo kresleno s využitím volně dostupného nástroje SMC eTool PneuDraw. Zvedání brusky válcem a přetlakování vedení os Festo ELGA jsou pneumatické prvky. Pneumatický obvod bude stavěný z aktuálních skladových zásob zákazníka, a proto byly některé komponenty navrženy pouze obecně, a navíc je pravděpodobné, že komponenty uvedené ve schématu budou nahrazeny za jiné podobné.

Přímočarý pneumotor, válec o průměru i délce 40 mm, je řízen třístavovým pěticestným ventilem s polohami spuštěno, zvednuto a zastaveno. Pneumotoru jsou v obou směrech předřazeny škrtkové ventily pro omezení rychlosti pohybu brusky. Vzhledem k malému objemu válce bude potřeba použít škrtkové ventily s velmi malým průtokem pro zamezení příliš rychlých pohybů a tím nárazů broušícího pásu do plechu či do koncových poloh. Proti nárazu v koncových polohách lze využít válce s tlumením v koncových polohách.

Prozatímní řešení přítlaku brusky na broušenou plochu je po spuštění brusky na plech zamýšleno přenechat na gravitačních silách. Tímto způsobem byl řešen přítlak i při manuálním provozu brusky. Obroušenost plechu je plánováno regulovat rychlostí posuvu. Pokud by však bylo obroušení nevyhovující, bude přistoupeno k modifikaci obvodu a práci s tlakem při broušícím procesu.

Vedení os Festo ELGA je předřazen tlakový ventil. Při přetlakování os totiž uniká část vzduchu z vnitřního prostoru vedení ven. Příliš velký tlak by mohl vyústit k velkému úniku vzduchu a provoz by byl drahý. Zároveň by mohlo dojít k ponížení pásu pružného plechu uzavírajícího vnitřní prostor. Tlak se tedy většinou volí nejvýše o 1 bar vyšší než okolní tlak, přičemž už přetlak 0,2 bar by měl být pro některé aplikace dostatečný. Konkrétní nastavení přetlaku je tedy předmět ladění až bude stroj hotový. *Informace nebylo možné nalézt v dokumentaci a byly získány při konzultaci s experty od Festo.*

Přívodní tlak do systému, resp. tlak dostupný ve výrobně zákazníka, je 6 bar. Tento tlak bude pravděpodobně snížen v úpravně vzduchu.



Obrázek 7.1: Schéma pneumatického obvodu

8 Řídicí systém

Softwarová část řízení stroje je rozdělena do dvou hlavních oddělených částí, které mezi sebou spolupracují. Jednou z nich je standardně program řídicího systému (IPC/PLC), který je zde zastoupen ve formě TwinCAT PLC runtime, starající se o chod a kontrolu celého systému řízení, dále má na starosti řízení a čtení aktuálních a sensorických komponent stroje, zobrazení uživatelského rozhraní (HMI) a kontrolu bezpečnosti stroje. Druhou částí je program vytvořený v rozhraní Matlab, založený na původní simulaci, který v reálném čase zpracovává data ze snímačů hlavy brusky a řídí její pohyby (data jsou čtena a zapisována z/do proměnných IPC). Skript z Matlabu je zkompileován do samostatné spustitelné aplikace a je spouštěn programem PLC.

8.1 Program PLC systému TwinCAT

Stavění programu je u průmyslových řídicích systémů upřednostňováno v jazyce Ladder diagram, protože jeho podobnost s elektronickými schématy jej činí dobře čitelný pro údržbáře a podobné zaměstnance výrobních závodů a ti si tak sami mohou analyzovat některé nastalé chyby a implementovat do kódu drobné změny bez toho, aby se specializovali přímo na programování PLC. Pro některé části programu je však i přes to výhodnější použít Strukturovaný text.

8.1.1 PLC runtime TwinCAT a ADS router

Řídicí software je provozován na průmyslovém počítači (IPC) Beckhoff. Výrobce pro svoje IPC poskytuje pokročilou řídicí platformu TwinCAT (The Windows Control and Automation Technology), kde se jedná o PLC runtime běžící na Windows. Jinými slovy je to vlastně virtuální PLC na počítači. Díky tomuto lze TwinCAT spouštět nejenom na Beckhoff IPC, ale také na jakémkoliv jiném běžném PC s OS Windows. To do jisté míry umožňuje vývoj programu bez přítomného PLC. TwinCAT (resp. TwinCAT XAE) je současně nazýváno vývojové prostředí na bázi Visual Studia od Microsoftu. TwinCATem je tedy nazýváno jako PLC, tak IDE, kde konkrétní význam je odvozen podle kontextu.

TwinCAT pro komunikaci mezi jednotlivými komponenty programu využívá rozhraní ADS routeru, který existuje na každém TwinCAT počítači. V praxi se jedná o rozšíření IP adresy o další dva byty. Komponentami využívajícími ADS router se myslí části programu jako PLC, NC osy, HMI či externí aplikace nebo zařízení na

EtherCAT síti. Pro využití ADS routeru externími aplikacemi Beckhoff poskytuje API ve formě knihoven pro různé platformy. Zde jsou využity například knihovny pro .NET, které umožňují Matlab skriptu číst a zapisovat proměnné PLC.

8.1.2 Struktura programu PLC

Program se skládá z obslužných podprogramů (POU) a z podprogramu starajícího se o pohyb brusky. Podprogram `_000_Main` obsahuje celý cyklus programu. `_002_Always` se stará především o periodické akce, jako vyčítání času nebo zdroje frekvencí. Podprogram `_006_EventLogger` udržuje systém událostí (alarmů). Pro ty TwinCAT nabízí řešení na bázi systému událostí z operačního systému Windows a nahrazuje zastaralé udržování událostí jako pole stringů. `_008_Devices` obsluhuje zařízení - zde je prováděno zapisování a čtení funkčních bloků fyzických zařízení jako pohony, nebo maják. Nakonec je zde podprogram `_010_Grinder` obsahující stavový automat ovládající stroj.

K bezpečnosti je vytvořen bezpečnostní podprojekt starající se o čtení a zapisování bezpečnostních vstupů a výstupů. Podprogram `_004_Safety` vyčítá stav bezpečnosti pro využití v programu a zajišťuje ovládání bezpečnosti skrze HMI. Pro řízení os i pneumatického ventilu byly využity firemní funkční bloky pro tyto účely.

8.1.3 Řízení servopohonů os

Pro řízení servopohonů byly využity funkční bloky (FB) z firemních knihoven využívající MC (Move Control) bloky z knihoven of Festa, resp. MC bloky pro obecné osy v HIL simulaci. Funkční bloky MC odpovídají standardu *PLCopen motion control*. Protože v aplikaci je předpokládán kontinuální pohyb os, kdy je měněna rychlost či cíl za jízdy a bez zastavení, na základě současných dat ze snímačů na hlavě brusky, bylo nutné upravit původní funkční blok pro řízení osy, protože FB MC standardně umožňují začít nový pohyb až po ukončení a zastavení posledního pohybu. Pro okamžitou změnu pohybu (rychlosti/cíle) či vytvoření fronty pohybů, je potřeba využít další instance funkčního bloku. Pokud je tedy vyžadováno změnit pohyb osy, či pohyb zařadit do fronty, je potřeba tento nový pohyb spustit z jiné instance funkčního bloku MC. Typy řazení/navázání pohybů jsou následující - okamžitě ukončit původní pohyb a bez zastavení začít nový a dále je zde zařazení pohybu do fronty, kdy se rozlišuje, zda osa nejdříve zastaví a poté začne nový pohyb, nebo nový pohyb začne v okamžiku, kdy by starý pohyb začal decelerovat. V této aplikaci byl využit pouze typ *MC_Aborting*, tedy první zmíněný. Toto nastavení je u bloků MC výchozí, a proto nebylo potřeba jej specifikovat přímo.

8.2 HMI - TwinCAT TF2000

Komunikace řídicího systému s uživatelem je zajištěna převážně skrze dotykovou obrazovku, kde je zobrazováno HMI také na platformě TwinCAT. Beckhoff pro TwinCAT nabízí dvě možnosti HMI - starší a jednodušší TF1800 a relativně nové TF2000,

kteřé, vzhledem k tomu, že je postavené kompletně na webových technologiích a editor ve TwinCAT IDE je vlastně jen specializovaný WYSIWYG editor, vytvářív vcelku mocný nástroj pro vizualizaci informací. Je zde umožněno vytvářet si vlastní funkce v jazyce JavaScript (či jQuery), vlastní komponenty, dále lze udržovat slovník výrazů pro snadnou tvorbu vícejazyčných obrazovek a lze hlídát téměř libovolnou událost (Event) a spouštět na ni akce. Projekt slouží částečně pro ověření využitelnosti této nové vizualizace, a proto je zde použita. Nevýhoda TF2000 oproti starší vizualizaci je vzhledem k větším možnostem vyšší náročnost na výkon IPC a na dovednosti programátora.

HMI se skládá z horní stavové/navigační lišty s tlačítkem pro zobrazení menu s odkazy na jednotlivé obrazovky a z rámce zobrazujícího aktuální obrazovku. Vizualně je rozhraní koncipováno, aby vyhovovalo firemním grafickým požadavkům.

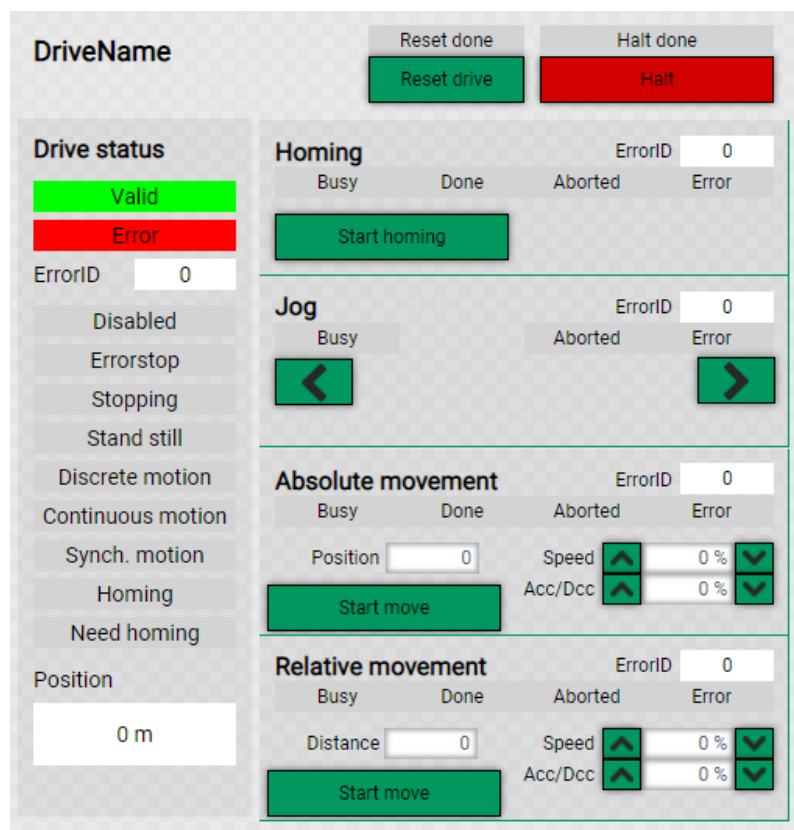
K HMI se připojuje jako k webové stránce přes prohlížeč a pro tento účel musí být v jeho konfiguraci vytvořen endpoint pro konkrétní IP adresu a port. HMI je spuštěno při startu Windows jako Chrome v kioskovém režimu a načtením adresy, kde je HMI publikováno.

Nové HMI však nenabízí pouze výhody, ale protože je stále spíše ve vývoji, obsahuje i drobné vady či nedodělky. Jedním z těchto nedostatků je implementace klávesnice, resp. způsob jakým lze v HMI zapisovat do vstupních polí. V nabídce jsou dvě možnosti. Jako první se nabízí využití virtuální klávesnice Windows. Zde však vznikl problém, kdy v desktopovém módu Windows virtuální klávesnici automaticky nezobrazují při aktivování vstupního pole a v tabletovém módu Windows klávesnice sice zobrazována je, nicméně po startu OS není umožněno spustit HMI na popředí Windows. Druhou možností je využití klávesnice dostupné v ovládacích prvcích pro tvorbu vizualizace. Řešení ze strany výrobce bývá ve formě, kdy je na okraji obrazovky tlačítko, kterým je klávesnice zobrazena. Bohužel zde se předpokládá, že klávesnice bude zaujímat statickou pozici a tím je omezen prostor, kam lze umístit vstupní pole, aby nebyla překryta klávesnicí. Protože toto je velice neefektivní způsob zacházení s prostorem, bylo přistoupeno k použití jQuery skriptu, který po zobrazení klávesnice umožňuje její libovolné přesouvání po obrazovce. Klávesnici je tedy vždy možné umístit tak, aby nepřekážela. Řešení pro tento zapeklitý problém bylo pochváleno přímo ze strany zastoupení výrobce systému.

Součástí tvorby HMI byla také tvorba některých ovládacích panelů pro kontrolu fyzických zařízení. Tyto panely byly po vzoru jiných projektů vytvořeny pro nové HMI. Jedná se o panel pro kompletní řízení servopohonů a panel pro ovládání pneumatických pohonů. Panel pro řízení servopohonů umožňuje kompletní řízení s využitím funkčních bloků podle standardu *PLCopen motion control*.

8.2.1 Obrazovka Přehled

Jedná se o výchozí obrazovku, kde je ovládán cyklus stroje a jsou zobrazeny základní údaje o jeho chodu. Nachází se zde ovládací tlačítka a pole pro zadání hranic plechu. Dále základní kontrola bezpečnosti.



Obrázek 8.1: Panel řízení servopohonu podle PLCopen

8.2.2 Obrazovky Manuál a Přehled stroje

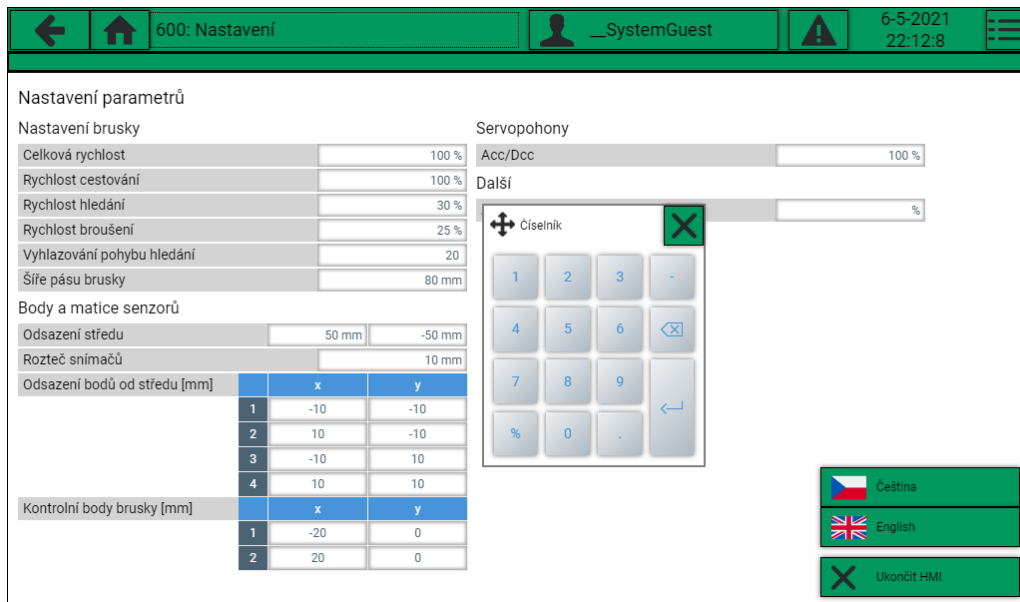
Tyto obrazovky slouží zejména pro manuální obsluhu, hledání poruchy a ladění programu. Z obrazovky *Manuál* se lze dostat na dvě podobrazovky obsahující kompletní manuální řízení servopohonů obou os, motorů brusky a odsávání a pneumatické zvedání brusky. Obrazovka *Přehled stroje* zobrazuje stavy všech snímačů a bezpečnostních prvků stroje.

8.2.3 Obrazovka Nastavení

Zde lze ladit veškeré parametry chodu programu. Jsou zde parametry jak programu PLC, tak jsou zde zadávány parametry, které si při spuštění vyčte program algoritmu. Dále je zde umožněna volba jazyku, či ukončení HMI pro návrat do Windows.

8.2.4 Obrazovky Info a Alarmy

Tyto obrazovky slouží zobrazení informací pro údržbu a hledání závad. Obrazovka *Info* zobrazuje informace o délce provozu motorů, statistické informace o provozu stroje a dále zmínku o integrátorovi. Obrazovka *Alarmy* obsahuje tabulku zobrazující události EventLoggeru, kde lze přepínat mezi zobrazením aktivních událostí a celou historií.



Obrázek 8.2: Obrazovka Nastavení

8.3 Zpracování dat s využitím Matlab

Pro zpracování dat a orientaci v prostoru byla použita původní simulace v prostředí Matlab, která byla upravena pro získávání dat z PLC skrze ADS router a uzpůsobena pro spojitý pohyb os. Pro tyto účely byly vytvořeny nové funkce, přičemž se veskrze jedná o funkce obstarávající čtení a zápis dat do PLC.

8.3.1 Převod na spojitý pohyb

V simulaci algoritmu se hlava brusky pohybuje nespojitě v pixelové mřížce testovacího obrázku plechu v rastrovém formátu (bitmapa). Změna směru pohybu je zde vždy pravouhlá, což by při použití se skutečným pohonem vyústilo v pomalý trhavý pohyb. Je tedy nasnadě pohyb vyhladit. Pro tento účel je pohyb řízen na základě váženého průměru historie předchozích pohybů a aktuálně žádané změny, kdy aktuálně žádaná změna má řádově vyšší váhu než předchozí pohyby. Váha je vedena jako měnitelný parametr a je předmětem ladění na reálném stroji. Tímto způsobem je docíleno změny směru, když je to potřeba a vyhlazení pohybu ústí v plynulejší změny poloh servopohonů a tím k rychlejšímu pohybu a vyšší životnosti stroje. V rámci zespojitého pohybu byla vytvořena funkce $normVec()$ převádějící vektor vyhlazeného pohybu na vektor o jednotkové velikosti - rychlost je s měnícím se směrem vždy stejné absolutní velikosti.

8.3.2 Komunikace s PLC

Základními kameny komunikace jsou funkce $readADS()$ a $writeADS()$ zjednodušující komunikaci za použití funkcí z originálních .NET knihoven. Funkce $connectADS()$

a *disconnectADS()* slouží pro připojení, resp. odpojení od PLC. Při spouštění programu jsou pomocí funkce *readParametersFromPLC()* načteny parametry programu podle hodnot zadaných v HMI.

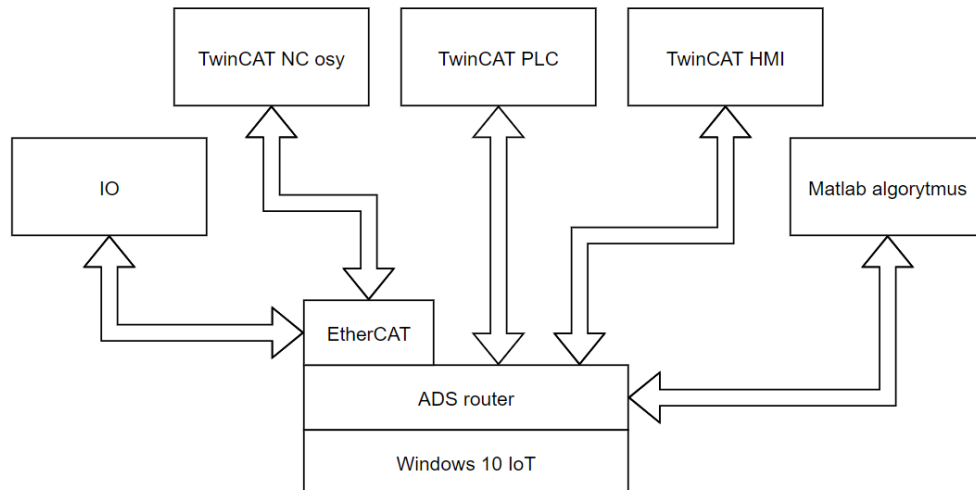
Pouhá změna souřadnic v proměnné v případě simulace zde byla nahrazena funkcemi pro zápis vypočtených směrových vektorů a pozic do proměnných PLC, podle kterých jsou řízeny servoměniče. Vytvořeno je zde několik funkcí pro rozdílné typy pohybů, kdy jsou implementovány pohybu v daném směru a pro přesunutí na absolutní souřadnice. S druhým typem pohybů se pojí funkce *waitForMoveToFinish()*, která slouží pro čekání až pohony dorazí na pozici.

8.3.3 Kompilace na samostatnou aplikaci

Protože není účelné, aby na IPC byl nainstalován celý Matlab, je skript zkompileován na samostatnou aplikaci. K tomuto je využito aplikace matlabu *Application Compiler*, jejímž výstupem je spustitelný soubor. Pro spuštění vytvořeného .exe souboru je také potřeba doinstalovat Matlab runtime. Instalace Matlab runtime je nevýhodou tohoto řešení, protože je vyžadována samotná instalace a runtime na IPC zabírá velké množství úložiště. Pokud se algoritmus na reálném stroji osvědčí, bude pro reálné nasazení zvažena jiná metoda chodu aplikace. Možnostmi jsou export funkcí Matlab skriptu do funkcí v jazyce C, pro podobu s jazykem Matlab a dostupnými knihovnamy je možností také přepsat program v jazyce Python nebo Octave.

8.4 Schéma systému jako celku

TwinCAT a jeho komponenty jsou založeny na vzájemné komunikaci skrze ADS router. Ten tak propojuje celý systém a vedle PLC runtime by se tak dal nazvat jeho středobodem. Algoritmus v Matlabu vyčítá z PLC stavy snímačů na hlavě brusky a na základě těchto dat vyhodnocuje následné pohyby - polohování servopohonů a zvedání brusky. Tyto informace po vyhodnocení zapíše do PLC. PLC pak navádí pohony na aktuálně vypočtené polohy.



Obrázek 8.3: Uspořádání řídicího systému

8.5 Simulace HIL

V době dokončování práce bohužel stále nebyla hardwarová část stroje dokončena ze strany zákazníka a kvůli koronavirové krizi, kdy je prakticky ve všech odvětvích průmyslu nedostatek komponent, byly výrazně prodlouženy termíny dodání některých zde využívaných komponent. Zejména se jedná o pohony Festo a panelové IPC Beckhoff, které jsou klíčovými komponenty řešení. Protože TwinCAT je spouštěn ve Windows a lze tedy PLC provozovat i na pracovním notebooku a zároveň je možné do TwinCAT projektu přidat simulované osy pohonů, bylo přistoupeno k ověření funkčnosti systému jako celku simulací *hardware in the loop (HIL)*.

Pohony byly přidány do projektu jako NC osy a byly napojeny na funkční blok pro kontrolu obecného pohonu (funkční blok referující pohon od Festo nemůže být použit pro simulaci). Během snahy zprovoznit simulaci byly odhaleny některé nedostatky v řízení běhu programu.

9 Závěr

Práce se zabývá návrhem a realizací řídicího systému pro automatické obroušení plechů. Plechy jsou broušeny kvůli povrchové korozi a znečištění vzniklých při skladování. Plechy mohou být ze skladu mírně prohnuté a zároveň jsou skladovány již obrobené s výřezy a dírami. Prohnutí a díry musely být brány v úvahu při návrhu řídicího systému.

V textu je čtenář průběžně seznámen s teorií broušení, broušeným materiálem a použitou bruskou. Práce pokračuje rozбором požadavků na algoritmus broušení a jeho návrhem a popisem. Následně je algoritmus otestován v prostředí Matlab. Po teoretické části následuje text zaměřený na výběr hardwarových komponent zařízení. Ten začíná provedením rešerše dostupných zařízení, dále je diskutována konstrukce a vybrány pohony. Pro stroj je navrženo zapojení elektrotechnických a pneumatických obvodů. Práce finišuje naprogramováním PLC, které však nebylo možné realizovat v plné míře, protože stroj zatím nebyl ze strany zadavatele postaven, a navíc dodání některých hlavních komponent je kvůli současné koronavirové krizi značně opožděno, a proto je funkčnost stroje testována s využitím simulačního modelu stroje.

V úvodní části věnující se broušení a brusce je ověřeno, že použitá bruska má již optimální rychlost a je rozhodnuto, že přítlak brusky na plech bude zajištěn pouze gravitačními silami a obroušení tak bude ovlivňováno rychlostí pohybu brusky po materiálu.

Kapitola o návrhu algoritmu nejdříve obecně zabývá nároky na algoritmus a popisuje různé přístupy k řešení této úlohy jak z pohledu sensorického, tak z pohledu koncepce algoritmu. Následuje popis samotného navrženého algoritmu, kdy k detekci plechu je využita matice čtyř snímačů umístěná na broušící hlavě, se kterou je pohybováno nad plechem a na základě údajů ze sensorů je zjišťován obrys plechu a rozhodováno o následujícím pohybu. Celkový cyklus broušení se skládá ze zadání přibližných polohových údajů o plechu operátorem, dále je podle těchto údajů hlava přemísťována nad plech. Přítomnost plechu je detekována senzory a tím je započato hledání vnějšího obrysu plechu. Obrys je hledán kvůli tomu, že operátorem zadaná poloha nemusí být přesná a dále pro zajištění obroušení celé plochy a vyhnutí se hranám, které by se mohly zničit nevhodným přebroušením. Obrys je nalezen a je s ním zacházeno jako s mnohoúhelníkem, protože díky převedení na tento matematický objekt lze ověřovat přítomnost vně či uvnitř zmapované oblasti. Po zmapování vnějšího obrysu je plech broušen, přičemž je zároveň hlídána přítomnost děr. Pokud jsou díky vyhodnocení dostatečně velké, je stejným způsobem i ze stejných důvodů jako u vnějšího obrysu zmapován i jejich obrys. Při návrhu algoritmu je ověřována jeho

funkčnost v prostředí Matlab na manuálně vytvořeném vzorku plechu. Nakonec je vytvořen generátor náhodných plechů, které jsou zdrojem speciálních případů pro hledání děr a na jeho základě byly z algoritmu odstraněny některé nedostatky.

Následuje část věnující se fyzickým komponentám systému, která je uvedena rešerší existujících strojů. Zde jsou představeny stroje podobného charakteru, nicméně nebyl nalezen žádný, který by dokázal automatickou detekci vyřezaného plechu a jeho následné obroušení. Následuje diskuse nad návrhem konstrukce stroje a jsou vybrány pohony. Stroj je navržen jako planární kartézský manipulátor a zvedání brusky je řešeno přes páku pneumatickým válcem. Osy manipulátoru jsou řešeny kompletně systémem od Festo s využitím přímočarých vedení ELGA se servopohony s motory EMMT a měniči CMMT. Osy jsou kompletně zakrytované a umožňují přetlakování vnitřních prostor vzduchem, což je činí odolnějšími proti zanesení kovovým prachem.

Elektronická část byla navrhována v elektrotechnickém dokumentačním software Eplan. Zde jsou uvedeny požadavky na řídicí systém a srovnáno několik jejich zástupců. Vybrán je automatizační systém založený na IPC od firmy Beckhoff Automation s.r.o., který byl vyhodnocen jako obecně nejlepší. Dále je popsáno elektrotechnické schéma, které je dostupné jako příloha a rozebráno dimenzování jeho komponent. V kapitole se nachází principiální blokové schéma uspořádání celého systému. Vedle návrhu elektroniky je navrženo a rozebráno schéma pneumatického obvodu. Zde je zmíněn použitý válec pro vertikální pohyb brusky a probráno nastavení přetlaku do přímočarých pohonů Festo ELGA.

V poslední části se text věnuje návrhu řídicího systému pro průmyslový počítač. Zde je vysvětlena technologie TwinCAT a ADS router následovaná strukturou programu. Výrobce automatizačního systému nabízí pro TwinCAT relativně nový systém vizualizace HMI. Jedním z cílů tohoto projektu bylo tento systém vyzkoušet a ověřit, a proto zde byl použit. Systém je založený na webových technologiích a blíží se klasické tvorbě webových stránek. Nabízí celou škálu pokročilých funkcionalit, ale zároveň má ještě nedostatky, se kterými se bylo potřeba vypořádat. Rozbor HMI následují stručné popisy obrazovek. Na tomto testovacím stroji vedle programu PLC paralelně běží program algoritmu. Původní skript simulace byl přetvořen, aby lépe vyhovoval nárokům na spojitý pohyb os a aby komunikoval s PLC. Program algoritmu běží na stejném IPC jako PLC runtime TwinCAT. Systém bohužel nebylo možné zprovoznit a ověřit jeho funkčnost na reálném zařízení, protože jej zadavatel stále nepostavil a některé klíčové komponenty systému mají kvůli současné koronavirové situaci velmi opožděné dodání. Kvůli nemožnosti ověřit funkčnost na reálném stroji byly do programu přidány simulační osy a byl proveden pokus o další simulaci, nyní jako celku s PLC, programem algoritmu a simulovanými senzory na pracovním počítači.

Závěrem rychlý sumář, čeho bylo docíleno. Byly provedeny potřebné rešerše a probrány potřebné předpoklady stroje. Byl navržen a otestován fungující algoritmus detekce plechu a broušení. Pro stroj byly navrženy elektrotechnické komponenty a vytvořeno elektrotechnické a pneumatické schéma. Byl navržen program řídicího systému a implementace algoritmu na IPC. Bohužel nebylo možné ověřit funkcionality na reálném stroji, a proto byla ověřena se simulovanými osami a senzory.

Literatura

- [1] Robert ČEP, Jana PETRŮ *TECHNOLOGIE OBRÁBĚNÍ: 5 BROUŠENÍ A DOKONČOVACÍ OPERACE OBRÁBĚNÍ* [online]. Ostrava, 2013 [cit. 2021-03-11]. ISBN 978-80-248-3012-4. Dostupné z: [http://projekty.fs.vsb.cz/463/edubase/VY_01_001/Technologie Obrábění/02 Text pro e-learning/Technologie obrabeni 05 Broušení a dokončovací operace obrábění.pdf](http://projekty.fs.vsb.cz/463/edubase/VY_01_001/Technologie%20Obrábění/02%20Text%20pro%20e-learning/Technologie%20obrabeni%2005%20Broušení%20a%20dokončovací%20operace%20obrabění.pdf)
- [2] JERSÁK, Jan. *Základní konvenční technologie obrábění: BROUŠENÍ BROUSICÍMI KOTOUČI* [online]. 2013 [cit. 2021-03-26]. Dostupné z: [http://educom.tul.cz/educom/inovace/TOB/VY_03_55-základní konvenční tchgie obrábění_BROUŠENÍ BROUSÍCÍMI KOTOUČI_MZ_6.pdf](http://educom.tul.cz/educom/inovace/TOB/VY_03_55-základní%20konvenční%20tchgie%20obrabění_BROUŠENÍ%20BROUSÍCÍMI%20KOTOUČI_MZ_6.pdf)
- [3] *Pásová bruska na kov GRIT GX 75. PEDDY* [online]. [cit. 2021-03-26]. Dostupné z: <https://www.peddy.cz/obrabeni-kovoobrabeci-stroje-brusky-pasove/-pasova-bruska-na-kov-grit-gx-75-223951>
- [4] *Inclusion of a Point in a Polygon* [online]. [cit. 2021-03-27]. Dostupné z: http://geomalgorithms.com/a03-_inclusion.html
- [5] XIA, Fan, Umme ZAKIA, Carlo MENON a Behraad BAHREYNI. *Improved Capacitive Proximity Detection for Conductive Objects through Target Profile Estimation* [online]. 2019 [cit. 2021-03-11]. Dostupné z: <https://www.hindawi.com/journals/js/2019/3891350/>
- [6] *ZTPolish: Stainless Steel Plate Mirror Polishing Machine For Carbon Steel Grinding Layer Removal* [online]. [cit. 2021-03-11]. Dostupné z: <http://www.automatic-polishingmachine.com/sale-11061455-stainless-steel-plate-mirror-polishing-machine-for-carbon-steel-grinding-layer-removal.html>
- [7] *Direct INDUSTRY: Metal polishing machine CDA-2000x12000/2UP-CP* [online]. [cit. 2021-03-11]. Dostupné z: <https://www.directindustry.com/prod/autopulit/product-16588-2219167.html>
- [8] *Direct INDUSTRY: Surface grinding machine BM* [online]. [cit. 2021-03-11]. Dostupné z: <https://www.directindustry.com/prod/cosmap-strl/product-67103-2179087.html>

- [9] *MATLAB Answers: Connect Matlab - TwinCAT trough ADS.dll, via ethernet* [online]. [cit. 2021-04-07]. Dostupné z: <https://www.mathworks.com/matlabcentral/answers/217553-connect-matlab-twincat-trough-ads-dll-via-ethernet>
- [10] *Beckhoff Information System: TwinCAT ADS* [online]. [cit. 2021-04-07]. Dostupné z: https://infosys.beckhoff.com/english.php?content=../content/1033/tcadscommon/html/tcadscommon_intro.htm&id=
- [11] Lishu Lv, Zhaohui Deng, Tao Liu, Zhongyang Li, Wei Liu, *Intelligent technology in grinding process driven by data: A review*, Journal of Manufacturing Processes, Volume 58, 2020, pp. 1039-1051, ISSN 1526-6125.

Přílohy

Seznam souborů na přiloženém CD

- MappingExecutable (spustitelný program algoritmu)
 - MappingForTwinCAT.exe
 - MyAppInstaller_web.exe
 - readme.txt
 - splash.png
- MappingMatlab (testovací skripty Matlab)
 - MappingForTwincat.m
 - MappingSim.m
 - MappingSimBatchTesting.m
 - MappingSimSmoothed.m
 - vzor plech - bg.png
 - vzor plech 4 points.txt
 - vzor plech 4.png
- PLC
 - Bruska_PLC.tnzip
 - Bruska_PLC_HMI.tnzip
- DP_Bruska_elektro_EPLAN.pdf
- DP_Bruska_pneu.pdf
- DP_Bruska_text.pdf
- DP_Bruska_text.tex