

Diagnostika provozu robotických pracovišť pro prediktivní údržbu

Diagnostics of Robotic Systems Operation for Predictive
Maintenance

Bc. Jan Zemánek

Diplomová práce

Vedoucí práce: prof. Ing. Jiří Koziorek, Ph.D.

Ostrava, 2023

Zadání diplomové práce

Student: **Bc. Jan Zemánek**

Studijní program: N0714A150001 Řídicí a informační systémy

Téma: Diagnostika provozu robotických pracovišť pro prediktivní údržbu
Diagnostics of Robotic Systems Operation for Predictive Maintenance

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Prediktivní údržba - základní principy, využití, metody. Rozbor možností využití prediktivní údržby u robotů a robotických manipulátorů.
2. Analýza možností sběru dat pro prediktivní údržbu z pohledu HW a SW.
3. Popis robotického pracoviště pro realizaci experimentu nasazení systému prediktivní údržby.
4. Systém sběru dat z robotického pracoviště, volba algoritmů a metod pro detekci a predikci stavu pracoviště.
5. Realizace měření a analýzy dat. Archivace a prezentace dat. Alarmní systém.
6. Zhodnocení výsledků řešení.

Seznam doporučené odborné literatury:

- [1] ČSN EN ISO 10218. *Roboty a robotická zařízení - Požadavky na bezpečnost průmyslových robotů*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2012. Třídící znak 186502.
- [2] Technická literatura firmy Staubli.
- [3] BOUCHARD, Samule. *Lean Robotics: A Guide to Making Robots Work in Your Factory*. [s.l.]:Samuel Bouchard, 2017. ISBN 1775082903.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Jiří Koziorek, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního programu: prof. Ing. Jiří Koziorek, Ph.D.

V IS EDISON zadáno: 16.11.2022 07:13:45

Abstrakt

Práce se zabývá problematikou monitorování robotických pracovišť a návrhem systému zpracování dat robotu za účelem aplikace prediktivní údržby. V práci je uvedena definice prediktivní údržby a její porovnání vzhledem k dalším typům údržby. Na základě metod a postupů konceptu prediktivní údržby je navržen systém sledování provozních dat robotu. Jeho účelem je detekce poruchy, nebo zhoršení mechanického stavu robotu. Součástí komplexního navrženého systému je sběr, záznam, analýza a archivace dat, včetně vizualizace výsledků analýzy. Vizualizace obsahuje integrovaný alarmový systém, který upozorňuje na změnu stavu robotu. V praktické části dochází k realizaci navrženého systému a integraci do robotického pracoviště. Funkčnost systému sledování stavu robotu je ověřena za pomoci simulace degradace. Ta je simulována pomocí závaží různých hmotností, které představují přidanou zátěž na pohony. Výsledkem práce je systém schopný v opakující se sekvenci robotu detekovat změny sledovaných provozních veličin a na základě algoritmů určit indikátor KPI reprezentující stav robotu.

Klíčová slova

Robot, sledování stavu, analýza dat, prediktivní údržba, klíčový ukazatel výkonnosti

Abstract

The thesis discusses the topic of monitoring robotic workstations and the design of a robot data processing system for the implementation of predictive maintenance. The definition of predictive maintenance and its comparison with other types of maintenance is introduced. Based on the methods and procedures of the predictive maintenance concept, a robot operation data monitoring system is designed. The purpose is to detect the failure or degradation of the mechanical condition of the robot. The complex designed system includes data collection, recording, analysis, and archiving, including visualization of the analysis results. The visualization includes an integrated alarm system that alerts the robot when the robot's condition changes. In the practical part, the proposed system is implemented and integrated into the robot workstation. The functionality of the robot health monitoring system is verified using degradation simulation. This is simulated by using weights of different masses to represent the added load on the actuators. As a result of the work, the system is capable of detecting changes of the monitored operating variables in a repetitive robot sequence and determine a KPI indicator representing the robot state by using algorithms.

Keywords

Robot, condition monitoring, data analysis, predictive maintenance, key performance indicator

Poděkování

Rád bych poděkoval vedoucímu diplomové práce prof. Ing. Jiřímu Koziorkovi, Ph.D. za odborné vedení této práce. Děkuji také Ing. Karlu Mikešovi za konzultace a cenné rady v průběhu realizace praktické části práce.

Obsah

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	7
SEZNAM ILUSTRACÍ	8
SEZNAM TABULEK.....	10
ÚVOD.....	11
1 ÚDRŽBA STROJŮ V PRŮMYSLOVÉM SEKTORU	12
1.1 Vývojové etapy údržby	12
1.2 Klasifikace údržby	13
1.2.1 Preventivní údržba	14
1.2.2 Korektivní údržba	14
1.2.3 Údržba s předem stanovenými intervaly	15
1.2.4 Údržba podle technického stavu	15
1.2.5 Aktivní údržba.....	15
1.2.6 Prediktivní údržba.....	16
1.3 Využití prediktivní údržby u robotů a robotických manipulátorů	17
1.3.1 Poruchy motorů.....	18
1.3.2 Poruchy robotů.....	19
1.4 Systémy monitoringu a PM od výrobců robotických manipulátorů	20
2 ANALÝZA MOŽNOSTÍ SBĚRU DAT	22
2.1 Sběr dat	22
2.1.1 Čtení dat z existujících zařízení.....	22
2.1.2 Externí senzorka	22
2.2 Hardware pro sběr, záznam a vyhodnocení dat.....	22
2.3 Software pro sběr dat.....	24
2.4 Přenos dat	26
3 POPIS ROBOTICKÉHO PRACOVIŠTĚ.....	29
3.1 Řídicí systém.....	30
3.2 Robotické manipulátory	31
3.2.1 Motorová výbava robotů.....	32
3.2.2 Předepsaná údržba robotů	33
3.2.3 Funkčnost robotu bez záložních baterií.....	34
3.2.4 Dostupné komunikační rozhraní robotu	35
3.2.5 Vnitřní proměnné robotu	35
3.2.6 Dostupnost proměnných v OPC UA.....	36
3.3 Komponenty externího měření proudů os robotu.....	37

4	NÁVRH SYSTÉMU VYHODNOCENÍ STAVU PRACOVÍŠTĚ A SBĚRU DAT	41
4.1	Analyzované veličiny	41
4.1.1	Proud	41
4.1.2	Točivý moment (Torque)	41
4.1.3	Teplota motorů.....	41
4.2	Algoritmy stavu pracoviště	42
4.3	Sběr dat	44
4.3.1	Robotická data.....	45
4.3.2	Řídící proměnné PLC.....	46
4.3.3	Měření proudu robotu, volba vhodného hardwaru	46
5	REALIZACE NAVRŽENÉHO SYSTÉMU MONITOROVÁNÍ ROBOTU	52
5.1	Sběr dat	52
5.1.1	Robotická data.....	53
5.1.2	Data z měřicích karet.....	55
5.2	Záznam dat do souboru.....	57
5.3	Zpracování dat	59
5.4	Uložení výsledků analýzy na SQL server	64
5.5	Vizualizace dat (Grafana).....	66
5.6	Realizace predikce stavu	71
5.7	Tok dat a automatizace procesu	72
5.8	Alarmové hlášení	74
6	OVĚŘENÍ FUNKČNOSTI SYSTÉMU	75
	ZÁVĚR	83
	LITERATURA	86
	SEZNAM PŘÍLOH	88

Seznam použitých symbolů a zkratek

AI	–	Analog input (analogové vstupy)
CSV	–	Comma-separated values (hodnoty oddělené čárkami)
ČSN	–	České technické normy
DB	–	Database (databáze)
DI	–	Digital input (digitální vstupy)
DO	–	Digital output (digitální výstupy)
FFT	–	Fast Fourier transform (rychlá Fourierova transformace)
GND	–	Ground (uzemnění)
HMI	–	Human machine interface (rozhraní mezi člověkem a strojem)
HS	–	High speed (vysokorychlostní)
HW	–	Hardware
I/O	–	Input/Outputs (vstupy/výstupy)
IOT	–	Internet of Things (internet věcí)
KPI	–	Key Process Indicators (klíčový ukazatel výkonnosti)
LAD	–	Ladder diagram (žebříčkový diagram)
MES	–	Manufacturing Execution Systems (výrobní informační systémy)
MS	–	Microsoft
OPC	–	OLE for Process Control
PC	–	Personal computer (osobní počítač)
PLC	–	Programmable logic controller (programovatelný logický automat)
PM	–	Predictive Maintenance (prediktivní údržba)
PXC	–	Phoenix Contact
RMS	–	Root Mean Square (Efektivní hodnota)
SCL	–	Structured Text (strukturovaný text)
SNPX	–	Series Ninety Protocol
SQL	–	Structured query language (standardizovaný strukturovaný dotazovací jazyk)
ST	–	Standart (standardní)
SW	–	Software
TCP/IP	–	Transmission Control Protocol/Internet Protocol

Seznam ilustrací

Obrázek 1 – Vývojové etapy údržby [4].....	13
Obrázek 2 – Klasifikace údržby dle normy ČSN EN 13306 [1], [2]	14
Obrázek 3 – Robotické pracoviště TQC testeru.....	29
Obrázek 4 – PLC Siemens ET 200SP.....	30
Obrázek 5 – Fanuc LR Mate 200iD	31
Obrázek 6 – Fanuc CR – 7iA/L.....	31
Obrázek 7 – Fanuc iPendant.....	32
Obrázek 10 – Připojení kabelu do robotu – harting konektor	37
Obrázek 11 – Připojení kabelu do kontroléru – rozdělení na více svazků.....	38
Obrázek 12 – Prodlužovací kabel k robotu.....	38
Obrázek 13 – Ukázka rozloženého harting konektoru	39
Obrázek 14 – Měřicí transformátory proudu	39
Obrázek 15 – Přenos proměnné proudu přes OPC při využívání Modbus proměnných	45
Obrázek 16 – Přenos proměnné proudu přes OPC při současné Modbus komunikaci	46
Obrázek 17 – Zapojení trafo Carlo.....	47
Obrázek 18 – Schéma zapojení serva Carlo [25]	47
Obrázek 19 – Zapojení trafo PXC.....	48
Obrázek 20 – Schéma zapojení trafo PXC [24]	48
Obrázek 21 – Nastavení přepínačů PXC karty	49
Obrázek 22 – Porovnání měřených průběhů trafo Carlo vs PXC na ST AI kartě.....	49
Obrázek 23 – Porovnání měřených průběhů trafo Carlo vs PXC na HS AI kartě	50
Obrázek 24 – Porovnání měřeného proudu s proudem získaným přes OPC.....	51
Obrázek 25 – Porovnání měřeného proudu a absolutní hodnoty proudu získaným přes OPC	51
Obrázek 26 – Porovnání aplikace smoothingu na analogové kartě PLC	51
Obrázek 27 – Blokové schéma systému sběru dat robotického pracoviště.....	52
Obrázek 28 – Detailní pohled na rychlost komunikace proudu osy přes OPC UA	53
Obrázek 29 – Program pro OPC UA komunikaci	54
Obrázek 30 – Ovlivnění proměnné Ring signalizující běh programu při komunikaci přes Modbus.....	55
Obrázek 31 – Naměřený průběh na jedné fázi 1. osy robotu za pomoci měřicího trafo PXC	56
Obrázek 32 – Program pro převod analogových hodnot proudu	56
Obrázek 33 – Definice S7-Explorer rozhraní v ibaPDA	57
Obrázek 34 – Sledování aktuálního průběhu zaznamenávání dat	57
Obrázek 35 – Definice zaznamenávaných proměnných v ibaPDA	58
Obrázek 36 – Prostředí ibaAnalyzer	60
Obrázek 37 – Analýza – Schéma analýzy dat pro jednu osu robotu	60
Obrázek 38 – Analýza – Načtení signálu proudu a rychlosti	61
Obrázek 39 – Analýza – Selekce analyzovaných průběhů.....	62
Obrázek 40 – Analýza – Výpočet statistických hodnocení průběhu	62
Obrázek 41 – Analýza – výpočet KPI statistických hodnocení.....	63
Obrázek 42 – Analýza – výpočet KPI osy robotu	64
Obrázek 43 – Běžící SQL Express server	64
Obrázek 44 – Automaticky generována struktura tabulek pro ukládání dat.....	65

Obrázek 45 – Definice připojení databáze	65
Obrázek 46 – Definice nahrávání signálů do databáze	66
Obrázek 47 – Ukázka tabulky deSegment_AvgT s nahranými signály	66
Obrázek 48 – Vizualizace – volba zobrazeného časového období	67
Obrázek 49 – Vizualizace – Volba uživatelských proměnných	68
Obrázek 50 – Vizualizace – obrazovka přehledu o stavu robotů	70
Obrázek 51 – Vizualizace – Obrazovka konkrétních informací o stavu robotu	71
Obrázek 52 – Aproximace průběhu za pomoci polynomu	72
Obrázek 53 – Blokové schéma toku dat systémem	72
Obrázek 54 – Ukázka prostředí aplikace ibaDatCoordinator	73
Obrázek 55 – Zobrazení panelu alarmů ve vizualizaci	74
Obrázek 56 – Ukázka zaslání alarmu na službu MS Teams	74
Obrázek 57 – Závaží pro simulaci zatížení robotu (2 a 4 kg)	75
Obrázek 58 – Zátěž připojená na efektor robotu	75
Obrázek 59 – Měření proudu při změnách zátěže robotu	76
Obrázek 60 – Stav zatížení robotu během testovacího měření	77
Obrázek 61 – Vliv zátěže na průměrnou hodnotu průběhu proudu 2. osy robotu	77
Obrázek 62 – Vliv zátěže na maximální hodnotu průběhu proudu 2. osy robotu	77
Obrázek 63 – Vliv zátěže na medián průběhu proudu 2. osy robotu	78
Obrázek 64 – Vliv zátěže na koeficient šikmosti průběhu proudu 2. osy robotu	78
Obrázek 65 – Vliv zátěže na koeficient špičatosti průběhu proudu 2. osy robotu	78
Obrázek 66 – KPI hodnocení průběhu proudu robotu na vybraných osách	79
Obrázek 67 – KPI hodnocení průběhu proudu robotu na 4. a 6. ose	79
Obrázek 68 – Celkové KPI hodnocení robotu	80
Obrázek 69 – Vliv zátěže na průměrnou hodnotu OPC průběhu proudu 1. osy robotu	81
Obrázek 70 – Vliv zátěže na průměrnou hodnotu měřeného průběhu proudu 1. osy robotu	81
Obrázek 71 – Vliv zátěže na maximální hodnotu OPC průběhu proudu 1. osy robotu	81
Obrázek 72 – Vliv zátěže na maximální hodnotu měřeného průběhu proudu 1. osy robotu	81
Obrázek 73 – Porovnání analýzy stavu osy robotu za pomoci měřeného proudu a proudu z OPC	82

Seznam tabulek

Tabulka 1 – Shrnutí pozitiv a negativ jednotlivých metod údržby [7]	17
Tabulka 2 – Struktura S7 komunikace [19].....	26
Tabulka 3 – Datové typy Modbus struktury	28
Tabulka 4 – Kódové označení funkcí Modbus komunikace.....	28
Tabulka 5 – Parametry analogových PLC karet [21], [22]	31
Tabulka 6 – Parametry robotů	32
Tabulka 7 – Typové označení a parametry motorů robotu Fanuc LR Mate 200iD.....	32
Tabulka 8 – Typové označení a parametry motorů robotu Fanuc CR – 7iA/L.....	33
Tabulka 9 – Modbus struktura proměnných robotu definována manuálem [23]	36
Tabulka 10 - Oprávnění pro čtení/zápis do Modbus struktury definována manuálem [23].....	37
Tabulka 11 – Parametry měřicích proudových transformátorů [24], [25]	40
Tabulka 12 – Nastavení přepínačů PXC karty	48
Tabulka 13 – Odesílané proměnné robotu přes OPC-UA.....	53
Tabulka 14 – Proměnné čtené z robotu za pomoci Modbus komunikace	54
Tabulka 15 – Proměnné odesílané do robotu za pomoci Modbus komunikace	54
Tabulka 16 – Nastavení analogové karty pro měření proudu robotu	56
Tabulka 17 – Seznam zaznamenávaných digitálních proměnných z PLC za pomoci ibaPDA	58
Tabulka 18 – Seznam zaznamenávaných analogových proměnných z PLC za pomoci ibaPDA	59
Tabulka 19 – Rozdělení časových značek v databázi.....	68
Tabulka 20 – Převod vzorkování mezi databází a vizualizací	69

Úvod

Diplomová práce se zabývá diagnostikou a sledováním stavu robotického pracoviště za účelem realizace prediktivní údržby robotu. Pojem prediktivní údržby je dnes velmi populární a nabízí systém údržby založený na moderních systémech zpracování dat. Cílem prediktivní údržby je predikce možné poruchy stroje a zamezení neplánovaným odstávkám. Na základě predikce stavu je umožněno plánování údržby nebo oprav a omezit dobu odstávky stroje. Funkčnost prediktivní údržby je založena na metodách, mezi které patří sledování provozních hodnot stroje, vibrodiagnostika nebo teplotní analýza. Na základě pozorovaného trendu analyzovaných dat lze provést predikci stavu nebo odhad, kdy dojde k dosažení kritické úrovně stavu stroje a je nutné provést servisní zásah.

Dle definovaného zadání se práce věnuje diagnostikou a návrhem systému sledování stavu robotického pracoviště. Zadání vychází z požadavků firmy Elvac a.s. Sledovaným objektem je robot Fanuc LR Mate 200iD. Cílem práce je návrh komplexního systému sledování provozních hodnot robotu, který dokáže detekovat změnu provozních veličin robotu v rámci prováděné sekvence. Systém monitorování se bude zabývat mechanickým stavem robotu a bude schopný rozpoznat změnu jeho provozních hodnot. Jakožto hlavními sledovanými komponenty budou motory, které slouží pro pohyb jednotlivých os robotu. Navržený systém bude obsahovat veškeré kroky práce s daty, od sběru dat, přes zpracování a archivaci, až po vizualizaci výsledků navrženého systému. Systém by měl fungovat automatizovaně a v určitých časových intervalech provést vyhodnocení stavu robotu za danou dobu. Výsledné hodnocení stavu by mělo ukazovat aktuální stav robotu a přehledně informovat obsluhu, zda došlo ke změně stavu.

První kapitola práce se zabývá základy údržby strojů v průmyslovém sektoru, vývojem údržby a možnými variantami typů údržby. Dále se věnuje konceptu prediktivní údržby a porovnává ji s dalšími typy údržby. Obsahuje informace o výhodách, nevýhodách a metodách, které je pro predikci stavu možné použít. Kapitola dále přechází na využití prediktivní údržby a sledování stavu stroje robotických manipulátorů. Je zde uvedeno, jaké poruchy se mohou na robotech vyskytnout a které metody jsou vhodné pro jejich predikci. V závěru kapitoly jsou uvedeny informace o systémech prediktivní údržby (condition monitoring), které nabízejí výrobci robotů Fanuc, KUKA a ABB.

Druhá kapitola se věnuje analýze možností sběru dat z robotického pracoviště a způsoby přenosu dat mezi zařízeními. Obsahuje popis použitých zařízení a postupů při sběru dat, ale také další varianty, o kterých bylo při realizaci uvažováno. Třetí kapitola obsahuje veškeré informace o vybavení robotického pracoviště, na kterém je práce realizována. Zabývá se parametry konkrétních prvků, které jsou pro výsledky práce podstatné a které jsou srovnávány v řadě prováděných testů. Součástí je popis komunikačních schopností, vlastností robotů a měřicích karet, pomocí kterých dochází k měření proudů na přívodním kabelu robotu.

Čtvrtá kapitola se zabývá návrhem analyzovaných veličin robotu a algoritmy, které jsou pro zpracování dat použity. Součástí je testování vhodného komunikačního rozhraní robotu, které lze pro čtení dat použít. Dochází zde také k popisu testování a výběru vhodné měřicí karty pro měření proudů na přívodním kabelu robotu.

Pátá a stěžejní kapitola se zabývá implementací navrženého systému do robotického pracoviště. Obsahuje popis funkčnosti jednotlivých částí navrženého systému analýzy stavu robotu. Jedná se o části sběru, záznamu, zpracování, archivace a vizualizace dat. Součástí vizualizace je také implementace systému predikce a alarmových hlášení. Poslední 6. kapitola popisuje způsob testování funkčnosti systému za pomoci simulace zhoršení stavu robotu a interpretuje jeho výsledky.

1 Údržba strojů v průmyslovém sektoru

Výroba za pomoci vysoce automatizovaných a robotizovaných strojů je hlavním pilířem dnešního průmyslu. Stroje jsou sestaveny z moderních a technicky složitých komponentů, které slouží pro zajištění správné funkčnosti, výkonnosti, udržitelnosti a spolehlivosti stroje. S provozem stroje je však pevně spjata jeho údržba a udržování stroje v provozuschopném stavu. Údržba je definována za pomoci ČSN EN 13306 [1] jako „kombinace všech technických, administrativních a manažerských zásahů během životního cyklu objektu zaměřených na jeho udržení ve stavu, v němž může vykonávat požadovanou funkci, nebo jeho navrácení do tohoto stavu.“

Často je údržba řazena mezi vedlejší aktivity podniku, má však přímý vliv na produktivitu výroby a zisk podniku. Především v dnešním velmi těsném konkurenčním boji je každé snížení nákladů na výrobu produktu vítané. To je s příchodem moderních metod, jako je monitorování a předpovídání stavu stroje, možné a nabízí úsporu nákladů na provoz stroje. V minulosti bylo hlavním cílem udržet stroj v bezpečném a provozuschopném stavu, kdy údržba nebo oprava byla často prováděna až na základě výskytu poruchy. Při poruše došlo za pomoci rychlé reakce k opravě postižené části stroje a co nejrychleji byl stroj uveden zpět do provozu. To z hlediska dostupnosti náhradních dílů nebylo vždy tak jednoduché a vznikly delší odstávky. Dnes převládá snaha o predikci možné poruchy a vyhnutí se poruše stroje, která by vedla k neplánované odstávce. Výroba je dnes často plánována v kooperaci s údržbou, a za pomoci využití určitých metod údržby lze přesně říci, kdy bude nutné údržbu na stroji provést. Tato kooperace mezi jednotlivými odděleními podniku vede ke zvýšení efektivity výroby a ke snížení nákladů. Podrobnější informace lze nalézt v [2], [3].

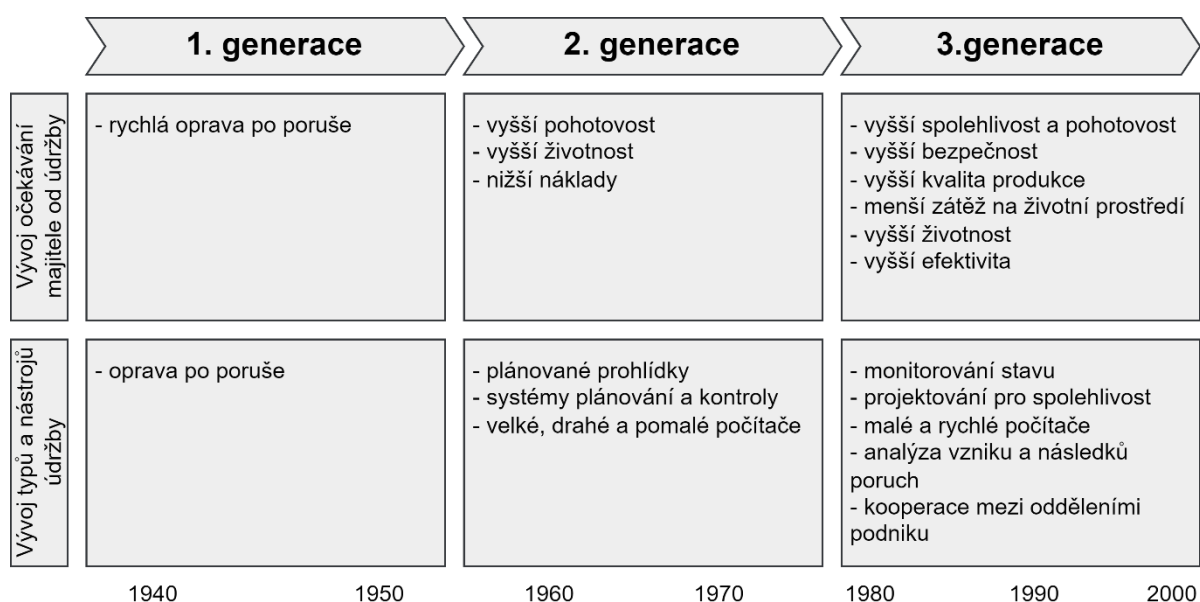
1.1 Vývojové etapy údržby

Historie údržby sahá až do dávné historie, kdy lidé začali vyrábět pomůcky a nástroje pro ulehčení jejich života. S příchodem nástrojů vznikla potřeba jejich údržby a opravy. Jeden z prvních záznamů o problémech s organizací údržby pochází z Libanonu roku 600 př. n. l. [4] a jsou v něm uvedeny problémy s dodávkami cedrového dřeva, které bylo potřeba na opravu lodi boha AmonRa. Údržba byla dlouhodobě opomíjena a oprava byla uskutečněna až při poruše lodi. Bez předem plánované opravy byla doprava dřeva zdlouhavá a opravu lodi vysoce prodražila. Ač nás od těchto problémů s opravou dělí více než 2 tisíciletí, s obdobnými problémy se zcela běžně můžeme setkat i v dnešní době. V historii lze nalézt i další příklady, kdy se člověk setkával s nástrahami údržby, ale velký rozvoj odvětví údržby nastal až s příchodem průmyslové revoluce. Průmyslová revoluce přinesla velký pokrok v oblasti výrobních procesů a mechanizace. S nástupem pokročilejších strojů a elektrifikace vznikla nová specializace pro odborníky na opravu strojů – profese údržbářů. Se vznikem pozice bylo nutné začít tuto činnost řídit, plánovat a organizovat.

John Moubrey definoval trojici vývojových generací údržby do dnešní doby (viz Obrázek 1), a to jak z pohledu očekávání od údržby, tak vývoje typů a nástrojů, které byly pro údržbu dostupné. První generace se datuje do 2. světové války, kdy průmysl nebyl příliš mechanizovaný, výpadky ve výrobě nebyly příliš řešeny a prevence poruchy stroje nebyla brána v potaz. Stroje a vybavení byly však jednoduché, což je dělalo spolehlivé a jednoduché na opravu. K velké změně došlo během 2. světové války, kdy nároky na spolehlivost a objem výroby drasticky vzrostl. Stroje začaly být více komplexní a došlo k velkému rozšíření mechanizace výroby. S rostoucími požadavky na průmyslovou výrobu došlo k většímu sledování a snaze zamezení neočekávaných výpadků v produkci. V této době vznikla preventivní údržba, která měla za cíl snížit pravděpodobnost výpadků v produkci za pomoci servisů

a údržby stroje. Náklady na údržbu začaly značně stoupat až se zavedením plánování údržby a snaze o co největší životnost stroje. Třetí generace přichází v období sedmdesátých let 20. století, díky rozvoji nových technologií a příchodu automatizace. Opět rostou nároky na vyšší spolehlivost, životnost, bezpečnost, kvalitu a nižší náklady na provádění údržby. Podrobněji v [4].

U vývoje typů a nástrojů údržby došlo k mezigeneračním pokrokům shodující se s vývojem očekávání. V první generaci stačilo opravit zařízení po poruše a žádné další postupy nebyly řešeny. V druhé generaci už docházelo ke zvýšení nároků na kvalitu a nákladů údržby, došlo tedy k zavedení preventivní údržby. Ve výrobě byl zaveden systém plánování údržby a začaly se využívat počítačové systémy. Ve třetí generaci došlo k výraznému pokroku v oblasti techniky, automatizace a počítačového řízení. Už při návrhu stroje bylo přemýšleno o způsobu, jakým bude stroj udržován a jeho konstrukce tomu byla přizpůsobena. Byly vyvinuty přístroje ke sledování stavu stroje a nové metody pro údržbu. Diagnostika stroje umožnila vznik a použití nových metod údržby podle stavu a prediktivní údržby, které jsou dnes využívány a pomáhají udržet náklady na údržbu co nejnižší.



Obrázek 1 – Vývojové etapy údržby [4]

1.2 Klasifikace údržby

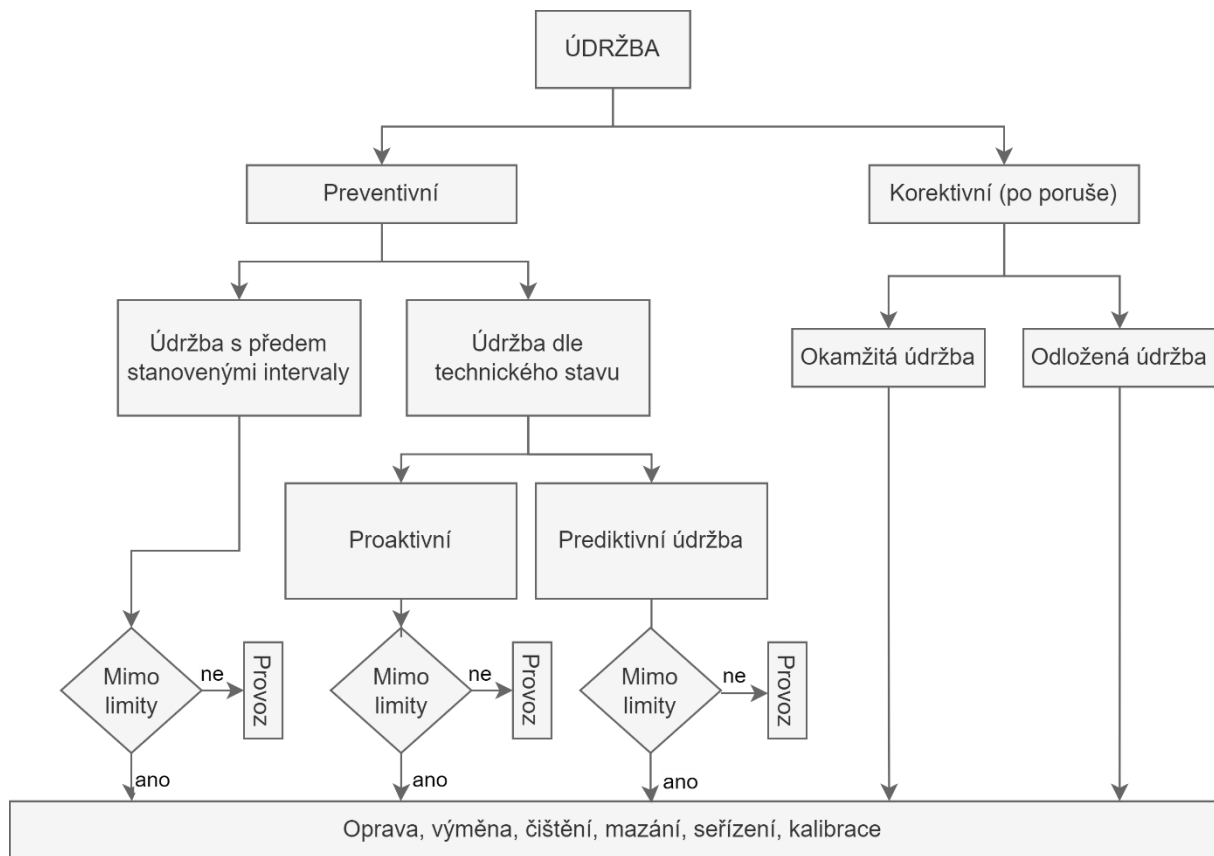
Jednotlivé přístupy pro provedení údržby lze rozdělit do několika kategorií a definují, kdy a za jakých podmínek dojde k provedení údržby na stroji. Za údržbu je považován zásah do stroje jako je oprava, výměna, čištění, mazání, seřizování, kalibrace, renovace, rekonstrukce a další.

Přístupy k údržbě vycházejí nejčastěji z:

- Stav stroje
- Stav komponentů stroje
- Časového období pro provedení údržby
- Jednotek používání stroje, či komponenty

Jednotlivé přístupy k údržbě lze rozdělit do dvou základních kategorií, a to na údržbu preventivní a korektivní. Jedná se o zcela opačné přístupy k vykonávání údržby. Zatímco preventivní údržba je prováděna z důvodu zamezení výskytu poruchy, korektivní údržba je prováděna až po poruše. Z pohledu provozu stroje ve výrobě se jedná o velmi zásadní kritérium, které je nutné vzít v potaz při

rozhodování, která z metod údržby bude použita. I přes toto rozdělení na údržbu preventivní a korektivní, dochází k dalšímu podrobnějšímu dělení (viz Obrázek 2). V následujících podkapitolách popisující dělení údržby je vycházeno z publikace Management a inženýrství údržby [2].



Obrázek 2 – Klasifikace údržby dle normy ČSN EN 13306 [1], [2]

1.2.1 Preventivní údržba

Preventivní údržba je dle ČSN EN 13306 definována jako „údržba prováděna se záměrem posoudit degradaci anebo zmírnit její následky a snížit pravděpodobnost poruchy objektu.“ [1] Základním principem preventivní údržby je provedení zásahu údržby před výskytem poruchy. Zásah spočívá v časně opravě či výměně komponentu stroje, která by mohla ohrozit jeho budoucí chod. Cílem preventivní údržby je udržet stroj v chodu a snížit riziko poruchy, která by mohla vést k zastavení stroje a omezení produkce. Preventivní údržba nabízí několik možných přístupů, jak lze údržbu vykonávat. Jednotlivé metody jsou rozděleny podle podmínek, za kterých je preventivní údržba provedena. Jedná se o časové intervaly, data z minulosti, nebo monitoring a ukazatele stavu stroje. Za výhody preventivní údržby lze považovat nízké riziko výpadku funkčnosti stroje, což je na druhou stranu spojeno se zvýšením nákladů ve spojitosti s nutností provádění pravidelných prohlídek a výměn komponentů.

1.2.2 Korektivní údržba

Korektivní údržba, taky označována jako údržba po poruše, je přesným opakem údržby preventivní. Dle ČSN EN 13306 je údržba definována jako, „údržba prováděna po zjištění poruchového stavu a zaměřená na obnovu objektu do stavu, ve kterém může vykonávat požadovanou funkci.“ [1] Jedná se o nejstarší typ údržby, který byl používán před nástupem moderních technologií schopných monitorovat stav a provoz stroje. Principem korektivní údržby je provedení zásahu až po výskytu poruchy na stroji. Údržbu lze dále rozdělit na okamžitou, kdy dojde k okamžité opravě, aby došlo

k zabránění nepříjemných následků, anebo odloženou, kdy je údržba odložena na základě určitých pravidel. Při výskytu poruchy a podle typu údržby, dojde k identifikaci poškozeného komponentu, následně k opravě či výměně a uvedení stroje zpět do provozu. Tento proces přichází s rizikem neočekávané a neplánované odstávky stroje po dobu jeho opravy. Je-li stroj v nepřetržitém provozu, může taková porucha ohrozit výrobní plán a způsobit dlouhodobou odstávku produkce. Jelikož k výměně komponentů dochází až po jejich selhání, je životnost komponentů maximálně využita a odpadá nutnost pravidelných kontrol a sledování stavu komponentů. Porucha jedné komponenty stroje však může způsobit další poškození stroje, které vznikne na základě nefunkčnosti porouchané komponenty. Vlivem většího poškození stroje pak dochází k prodloužení doby a úsilí nutné na opravu.

1.2.3 Údržba s předem stanovenými intervaly

Dle normy ČSN EN 13306 se jedná o „preventivní údržbu prováděnou v souladu se stanovenými časovými intervaly, nebo stanoveným počtem jednotek používání, avšak bez předchozího zkoumání stavu objektu.“ [1] Údržbu lze pojmenovat také jako predeterminovanou a je prováděna na základě časových úseků, které jsou předem definovány za pomoci předchozích znalostí, nebo za pomoci doporučených intervalů pro údržbu. Intervaly jsou určeny na základě časového rozmezí nebo jednotkou používání stroje. Pravidelná kontrola a údržba stroje zaručuje nízké riziko vzniku poruchy, rostou však náklady na provoz.

1.2.4 Údržba podle technického stavu

Norma ČSN EN 13306 definuje údržbu jako „preventivní údržbu, která zahrnuje posouzení fyzického stavu, analýzu a případné následné zásahy údržby.“ [1] Cílem této metody je včasné odhalit selhávající, či degradující stav zařízení, a to za pomoci kombinace monitorovacích technik, analýz, testování, sledování fyzického stavu nebo odchylek od normálního stavu. Stav zařízení je možné kontrolovat během fyzických kontrol, které posuzují stav stroje a definují nutné práce na údržbě. Stav zařízení je možné také sledovat za pomoci monitorovacích systémů, které poskytují informace o provozu zařízení. Monitorování je realizováno za pomoci sledování vnitřních veličin zařízení, nebo za pomoci specificky umístěných monitorovacích zařízení, které poskytují žádané hodnoty o sledované veličině. Veličiny jako jsou teplota a vlhkost, ve kterém je zařízení provozováno, mohou také poskytovat důležité informace, které pomáhají k určení stavu zařízení. Stav zařízení či určité části stroje, je obvykle vyjadřován za pomoci indikátoru zdraví neboli Key Process Indicators (KPI). Při zhoršujícím stavu zařízení dochází k postupné změně celkového KPI zařízení a obsluha jednoduše rozpozná změnu ve stavu zařízení. KPI je vyjadřováno v procentech, kdy 100 % značí zařízení ve zdravém stavu a poklesem směrem k nule dochází ke zhoršování stavu.

1.2.5 Aktivní údržba

Norma ČSN EN 13306 definuje aktivní údržbu jako „údržbu, ve které jsou zásahy prováděny přímo na objektu tak, aby byl držen ve stavu, ve kterém může vykonávat požadovanou funkci, nebo byl do něho navrácen.“ [1] V některých literaturách označována také jako proaktivní údržba. Jedná se o pokročilou metodu údržby, která spojuje preventivní údržbu s údržbou prediktivní. Úkolem aktivní údržby je zajištění plánování opravy, výměny, a to před koncem životnosti komponentů stroje, bez kladení důrazu na skutečný technický stav. Na základě degradace rozpoznané pomocí monitorování stavu stroje, dochází k preventivní výměně degradované komponenty [5].

1.2.6 Prediktivní údržba

Dle normy ČSN EN 13306 je definována jako „údržba podle stavu prováděna na základě předpovědi odvození z opakované analýzy, nebo ze známých charakteristik a vyhodnocení parametrů degradace objektu.“ [1] Cílem prediktivní údržby je za pomoci analytických a statistických metod analyzovat a vyhodnotit data, která vedou k určení stavu stroje. Na základě informací o aktuálním a předešlém stavu stroje, dochází k předpovědi budoucího vývoje stavu stroje a určení možného času nutné údržby. Dle predikce je možné předem naplánovat údržbu stroje, nebo výměnu vadných komponentů [6]. Výhoda je znatelnější při nutnosti výměny komponentu se špatnou dostupností. Pro získávání dat o stroji jsou využívány monitorovací systémy, které za pomoci senzorů stroj monitorují a poskytují hodnoty na vyhodnocení. Za pomoci analýz probíhá zpracování dat s výstupem ve formě grafů a indikátorů, které jsou dostupné v určité vizualizační formě pro obsluhu stroje. Součástí systému mohou být zakomponovány alarmy, které indikují překročení nastavené meze stroje a umožňují odeslání emailu nebo jiného způsobu informování o nastalé situaci.

V některé literatuře je prediktivní údržba a údržba podle technického stavu vnímána jako shodný pojem. Obě metody se shodují v nutnosti monitorování stavu stroje, avšak prediktivní údržba slouží také pro predikci budoucího stavu stroje, na rozdíl od metody údržby podle technického stavu, která definuje pouze aktuální stav [7].

Prediktivní údržba nabízí několik technik, na základě kterých dochází k definování stavu stroje [8]. Tyto techniky určují, z kterých provozních veličin, nebo z kterých dat bude predikován jeho stav. Technik je možné definovat velké množství, příkladem lze použít monitoring provozních veličin, vibrací, teplot, hluku, tribologie, nebo ultrazvukové analýzy. Z velké většiny se jedná o neinvazivní techniky, které nemají žádný vliv na sledovaný stroj a neovlivňují tedy jeho chod.

Monitorování výkonnosti a provozních hodnot

Jednou z rozšířených technik je monitorování provozních hodnot stroje a s tím související výkon stroje. Hodnoty mohou být dostupné již v rámci samostatného stroje, nebo je nutno rozšířit stroj o zařízení, které potřebnou veličinu bude měřit. K hodnotám, které lze vyčíst ze stroje, lze často přistupovat přímo z řídicího zařízení stroje (PLC), tudíž jsou dobře dostupná a je možné je lehce předat do vyhodnocovacího členu. Při vybavení stroje dodatečnými senzory pro měření žádané veličiny, je nutné tyto hodnoty dostat do řídicího členu stroje, případně jiným způsobem předat do vyhodnocovacího členu, což může být technicky náročné a nákladné. Příkladem provozních veličin může být proud, napětí, elektrická kapacita, elektrická indukce, odpor a další [9].

Monitoring vibrací

Průmyslové stroje jsou z většiny konstruovány z mechanických a pohyblivých částí, u kterých vznikají vibrace. Sledováním intenzity a změn vibrací stroje lze monitorovat změny v chování jeho komponentů a odhalit možné problémy. Vibrační senzory jsou umísťovány na sledované mechanické části stroje a při chodu stroje poskytují informace o tom, jak moc dochází v různých stavech stroje k jeho vibracím. V posledních letech jsou používány senzory na bázi mikroprocesoru s jednonálovým sběrem dat a vyhodnocením za pomoci aplikace na klasickém Windows PC. S rozšiřující popularitou sledování stavu strojů a vibrací stroje došlo k tvorbě hardwaru také ze strany velkých společností jako je Siemens, který nabízí platformu SIPLUS CMS jako rozšiřující modul k PLC a umožňuje přímé připojení vibračních senzorů. K vyhodnocení vibračních průběhů stroje se využívá FFT a následná identifikace frekvencí s největším výskytem. Pomocí analýzy lze například u motoru detekovat poruchu na vinutí, statoru,

rotoru, nebo jiných částech. I přes použití vibrodiagnostiky je nutné použití dodatečných metod a zpracování dalších doplňujících provozních veličin pro určení správného stavu stroje.

Monitoring teploty

Monitorování teploty stroje spočívá ve sledování teplot a analýze, zda nedochází k nadměrnému zahřívání ve sledovaném místě. Určité stroje mohou mít vlastní integrované teplotní senzory, pomocí kterých lze teplotu sledovat, jedná se však o jednu teplotu, kterou nelze porovnávat s dalšími teplotami okolních součástí, které již teplotním senzorem vybaveny nejsou. K tomu se využívají infračervené kamery, které směřují na určitou část stroje a detekují energii, kterou povrch vyzařuje. Tím dojde k určení teploty povrchu v několika místech zároveň a lze porovnat části stroje mezi sebou. Pomocí detekce oblastí, které jsou buďto teplejší, nebo studenější, nežli by měli být, je možné detekovat konkrétní místo problému. K detekci se používají infračervené teploměry, řádkové skenery, nebo infračervené termografy.

Vizuální inspekce

Vizuální inspekce patří mezi základní a historicky nejstarší metody pro sledování stavu stroje. Již od začátků průmyslové revoluce docházelo k prohlídkám stroje za pomoci lidské obsluhy a detekce možných anomálií v chodu stroje. Technik sledoval pohyby stroje, slyšel zvuk stroje, mohl ověřit vibrace stroje. Tato metoda je zachována do dnešních dnů a je součástí denních kontrol strojů. Při využití pokročilých metod sledování stavu stroje se však jedná pouze o dodatečnou kontrolu.

Jak vyplývá z předchozího textu, metod údržby je velké množství a každá přichází se specifickými klady, ale také zápory použití. Většina metod se snaží nalézt ideální poměr mezi bezproblémovým chodem stroje a co možná nejnižšími náklady. Jednotlivé pozitivní i negativní stránky metod jsou shrnuty v následující tabulce (viz Tabulka 1).

Tabulka 1 – Shrnutí pozitiv a negativ jednotlivých metod údržby [7]

Metoda údržby	Pozitiva	Negativa
Korektivní	<ul style="list-style-type: none"> Maximální využití životnosti komponentů Odpadá nutnost sledování stavu stroje 	<ul style="list-style-type: none"> K poruše stroje dochází v nepředpověditelném čase Porucha komponentů může způsobit rozsáhlejší poškození Doba odstávky stroje je závislá na rychlosti opravy
Preventivní – Údržba s předem stanovenými intervaly	<ul style="list-style-type: none"> Prodlužuje životnost zařízení Umožňuje naplánování údržby 	<ul style="list-style-type: none"> Vyšší náklady z důvodu provádění častější údržby Vyšší náklady na měněné komponenty, které nemusí být na konci své životnosti Nutnosti správného určení intervalu údržby
Preventivní - Údržba podle stavu	<ul style="list-style-type: none"> Znalost aktuálního stavu stroje Umožňuje údržbu plánovat na základě stavu stroje 	<ul style="list-style-type: none"> Nutné náklady na vybavení stroje diagnostickým systémem pro monitorování stavu (HS, SW) Dodatečné náklady na provoz monitorovacího zařízení
Preventivní - Prediktivní údržba	<ul style="list-style-type: none"> Znalost aktuálního stavu stroje Dokáže predikovat zhoršující se stav stroje a předpovídat jeho poruchu Porucha je předpovězena a je možné naplánovat údržbu 	<ul style="list-style-type: none"> Nutné náklady na vybavení stroje diagnostickým systémem pro monitorování stavu (HW, SW) Náročný vývoj prediktivního modelu pro konkrétní stroj Možná nepřesnost prediktivního modelu

1.3 Využití prediktivní údržby u robotů a robotických manipulátorů

Prediktivní údržba má své místo u všech pohyblivých, mechanických, nebo rotujících zařízení, u kterých může dojít k opotřebování, vlivu prostředí na chod, nebo jinému riziku poruchy zařízení. Velmi žádanou

kategorií pro provádění prediktivní údržby jsou motory. Motory obsahují pohyblivé části, ložiska, vinutí a další části, u kterých může dojít k poškození nebo opotřebením.

Vhodné veličiny a metody pro analýzu stavu motoru [8]:

- Vibrace
- Okolní teplota
- Zvuková kontrola
- Termografie
- Proud motoru
- Izolační odpor
- Induktance

Danými metodami lze analyzovat stav motoru a sledovat, zda nedochází k jeho zhoršování. Obdobným způsobem lze analyzovat provozní data robotického manipulátoru. Robot je vybaven motory na svých jednotlivých osách a za pomoci vnitřních proměnných robotu sledovat jejich stav. Na základě toho lze pozorovat, zda nedochází k změně sledovaných průběhů, nárustu potřebného výkonu při stejné operaci, nebo jiné změně měřeného průběhu v čase. Prediktivní údržba u motorů umožňuje včasnou detekci zhoršujícího se stavu motoru a dokáže detekovat mechanické i elektrické poruchy před způsobením větších škod. To přináší následující výhody [10]:

- Zvýšení dostupnosti stroje ve výrobě, snížení času výpadků kvůli údržbě
- Snížení nákladů na opravu po poruše stroje
- Možnost přesného plánování provedení údržby stroje
- Detekce výkyvů provozních hodnot díky monitorování stroje
- Historické trendy chování stroje
- Sběr dat o chování stroje
- Minimalizace neočekávaných servisních zásahů
- Umožňuje vzdálený monitoring stroje a zasílání dat do nadřazeného systému

1.3.1 Poruchy motorů

U motorů lze definovat základní trojici komponentů, u kterých dochází k poruše. Tyto poruchy mohou být za pomoci určitých metod monitorování stavu detekovány a předpovídaný před výskytem. Mezi komponenty motoru náchylné k poruše patří [11]:

Porucha ložiska – může být způsobena špatnou konstrukcí, mechanickým namáháním, špatným umístěním při výrobě nebo špatnou lubrikací. Porucha může mít vliv na všechny části motoru.

Porucha statorového vinutí – způsobena přehřátím, zkrácenou cívkou, špatnou konstrukcí nebo kontaminací. Při poruše dochází k nerovnováze v cívce způsobující vibrace.

Porucha rotoru – často způsobena špatnou rovnováhou rotoru, nesouměrností, lomem.

Porucha může nastat také ve spojce, ose, nebo v převodech motoru a vzniká nejčastěji na základě nadměrného zatížení motoru. Motory, které jsou umístěny ve venkovních a náročnějších podmínkách jsou více náchylné na poruchu, než motory umístěny ve vnitřním prostředí se stabilními podmínkami. Také vysokonapěťové a vysokootáčkové motory jsou mnohem více náchylné na poruchu než nízkonapěťové a nízkootáčkové motory [12]. Poruchy motorů mohou být způsobeny mnoha faktory a mnoha veličinami, které působí na chod motoru.

Jedná se o následující faktory [11]:

Prostředí

- Teplota
- Vlhkost
- Odvětrávání
- Znečištění
- Rez

Lidský faktor

- Špatný výběr typu motoru
- Špatný způsob použití
- Nedostatek údržby
- Nevhodná údržba

Provoz

- Vibrace
- Přetížení
- Časté starty

Vybavení

- Stáří
- Kvalita
- Návrhové chyby
- Chyby při výrobě

Napájení

- Přechodové jevy způsobené zkratem, výkyvy rezonancí, izolací, řízením, kondenzátory
- Vadná elektroinstalace
- Pokles napětí
- Nízké napětí

1.3.2 Poruchy robotů

Zatímco motory jsou obecně užívány pro neustálé otáčení, případně polohování na základě enkodéru, servo motory jsou určeny především pro přesné polohování. Servomotor není konstruován na nepřetržitě otáčení, ale na pootočení o danou vzdálenost nebo otočení o daný počet otáček. Právě servomotory jsou využity ve většině robotických manipulátorů jako pohony pro pohyb jednotlivých os. Stejně jako točivé třífázové motory obsahuje servomotor pohyblivé části a komponenty, které jsou náchylné na opotřebení nebo poškození. Jedná se o opotřebení ložiska, vady rotoru, brzdy, převodů servomotoru nebo jiných částí. Servomotor tedy lze obdobně jako klasický motor monitorovat a sledovat jeho stav za pomoci analýz napájecích veličin, vibrodiagnostiky, termografie apod. Z provedené rešerše jsou uvedeny nejčastější poruchy servomotorů [13], [14]:

- Kontaminace
- Ložiska
- Degradace elektronických obvodů
- Selhání vinutí
- Vada brzdy
- Poškození nesprávnou instalací a použitím

Při komunikaci s výrobcí robotických manipulátorů jsem se tázal na komponenty, které jsou na robotech nejvíce poruchové, nebo u kterých dochází k pravidelným výměnám, potažmo servisování. Napříč výrobci (Fanuc, KUKA, ABB) jsem obdržel obdobné informace. Nelze přesně říci u kterých komponentů dochází k nejčastějšímu poškození nebo podléhají nízké výdrži. Celkově jsou robotické manipulátory dimenzovány na několikaletý provoz bez výskytu větší poruchy. Nejčastějším důvodem k poškození určité části jsou kolize a nesprávný provoz robotu. Při vzniku kolize sice dojde k zastavení robotu, ale až v reakci na nadproud v jednotlivých osách. Při kolizi tedy dojde ke kontaktu s okolními překážkami, a to může způsobit poškození některé z částí robotu. Při opakovaných kolizích, například jednou osou robotu, může dojít k poškození převodovky, přetrhnutí řídicích kabelů, nebo jiného

vnitřního poškození. Z hlediska nedodržování předepsané údržby k poškození dojít může, ale dle dat výrobců se jedná pouze o ojedinělé případy a většina poškození vzniká na základě kolizí robotu.

1.4 Systémy monitoringu a PM od výrobců robotických manipulátorů

Práce se zabývá návrhem systému monitorování stavu robotu na základě kterého lze predikovat jeho vývoj. I když se jedná o poměrně populární oblast z hlediska využívání zpracování dat a užití IOT senzorů v průmyslu, produkty pro funkční predikci stavu jsou stále v začátcích vývoje. Během realizace práce jsem kontaktoval trojici největších výrobců průmyslových robotů, a to firmy Fanuc, KUKA a ABB. Kontaktoval jsem je s žádostí o poskytnutí informací o jejich systémech sběru dat, monitoringu a prediktivní údržbě. Na webových stránkách společností byly dostupné určité informace, ale často neuváděly informace ani o základních principech funkčnosti monitorovacího systému. Některé z firem byly sdílnější a ochotnější poskytnout informace o funkčnosti jejich systému, ale některé tyto informace vnímali jako velmi citlivé, a kromě veřejně dostupných informací jsem se nedozvěděl nic navíc. Nutno zmínit, že kontaktovány byly vždy české pobočky firem. Kladené otázky se týkaly funkčnosti systému monitoringu, poruchových částí a příčin vzniku poruch.

Fanuc

Jelikož je diplomová práce realizována na robotu Fanuc, první oslovenou firmou byl samotný Fanuc. Následující informace vyplývají z obdržených informací, nebo z jejich webových stránek. Fanuc ve svém portfoliu nabízí software Zero Down Time, který slouží pro monitorování stavu a údržby robotů. Software ZDT je nasazen ve firemní síti a roboty na něj ukládají veškerá provozní data. Variantou provozování systému je také umístění serveru externě a připojení více sběrných PC z různých firem, které na něho budou data zasílat. Systém monitoruje následující části robotu:

- Stav mechanických částí – systém umožňuje sledování neobvyklého chování servomotorů, převodů a zajistit jejich včasnou údržbu při zhoršení stavu. Příkladem je uváděno sledování Torque (točivého momentu) motorů.
- Stav údržby robotu – sledování doby pro dodržení předepsané údržby a sledování doby předpokládané výdrže komponentů v robotu
- Stav systému – monitorování stavu baterií v robotu, úroveň volného místa na disku apod.
- Stav procesu – sledování procesu svařování, bodového svařování

Tento nástroj pracuje v kooperaci s řídicí jednotkou robotu, ve které dochází k výpočtu algoritmů pro určení stavu robotu. I přes přímé kontaktování Fanucu nelze s větší přesností říci, které veličiny robotu jsou analyzované anebo jak vypadá algoritmus výpočtu stavu. Vše se odehrává uvnitř kontroléru a bez možnosti se k datům jakkoliv dostat. Výstupem z kontroléru je již výsledek provedené analýzy. Lze předpokládat, že uvnitř kontroléru dochází k výpočtu z proměnných, které nejsou přístupné ani v robotu jako takovém a dokáže k nim přistupovat pouze nástroj Fanucu. Při komunikaci mi bylo sděleno, že vývoj těchto nástrojů je alokovan přímo v domovském Japonsku a nikdo jiný přesnější informace o vnitřním fungování algoritmů nezná. Pro funkčnost zmiňovaného systému není potřeba žádných dodatečných zařízení a vše potřebné je již v řídicí jednotce robotu.

ABB

Jako další byl kontaktován výrobce robotů ABB, který ve svém portfoliu nabízí přímo službu prediktivní údržby. Ta je složena z nepřetržitého monitorování robotů a odesílání dat na uložiště, v kombinaci se zákaznickým servisem, který reaguje na vzniklé alarmy/nestandardní stavy při analýze dat. Zdrojem dat

je řídicí jednotka robotu, která odesílá provozní data robotu na server/uložiště, na kterém dochází k jejich dlouhodobému ukládání. Na základě definovaných pravidel a algoritmů dochází ke zpracování dat a vyhodnocení stavu robotu. Z pohledu mechanického poškození jsou analyzovány a sledovány proudy motorů, napětí na řídicích obvodech, pohyb robotu, rychlost, kroutící moment nebo akcelerace. Součástí nástroje prediktivní analýzy je SIS analýza (Condition Based Maintenance), která pomáhá s předpovědí zbývající životnosti převodovek. ABB u tohoto nástroje uvádí úspěšnost až 80 % a cílem analýzy je odhalení blížícího selhání nebo zhoršení stavu převodovky. SIS analýza vychází z globálních dat všech ABB robotů daného typu a na základě jejich průběhů dokáže rozpoznat nestandardní chování. Obdobně jako u ostatních výrobců je vývoj nástrojů a algoritmů pro prediktivní údržbu situován do domovského ABB, a kromě úzkého okruhu vývojářů je přesná definice algoritmů neznámá. Mezi další sledované parametry, které se netýkají přímo mechanického stavu robotu patří napětí na záložních bateriích, stav uložště robotu, stav RAM paměti nebo monitoring systému chlazení. Pokud pomocí algoritmu dojde k vyhodnocení nestandardní situace, dojde k vygenerování alarmu, který je předán týmu analytiků. Jejich úkolem je analyzovat daný alarmový stav a za pomoci dostupné historie stavu robotu rozpoznat, zda nedošlo k poškození, zhoršení stavu nebo jinak ovlivnění provozuschopnosti robotu. Příkladem analyzovaného alarmu je například nouzové zastavení v důsledku kolize, nadměrné zatížení jedné z os, poklesu napětí na bateriích nebo jiné pozorované události. V takovém případě dojde k analýze dané situace a realizace servisních zásahů, opravy nebo upozornění obsluhy na danou situaci.

KUKA

Poslední z kontaktovaných výrobců robotů je firma KUKA. Na úvod je nutno zmínit, že na základě vzájemné komunikace nebyly poskytnuty žádné informace, jelikož žádané informace o funkčnosti systému prediktivní údržby nejsou dle jejich vyjádření veřejně šiřitelné. Veškeré informace jsou tedy čerpány pouze z webových stránek KUKA [15] a jsou tedy značně omezeny. Nabízený nástroj pro účely monitorování stavu robotů je služba KUKA iiQoT. Jedná se o nástroj shromažďující data z jednotlivých robotů a integraci jejich stavu do jednoho prostředí. Funkčnost systému iiQoT je rozdělena do modulů, které se zabývají konkrétní oblastí:

- Asset manager – získávání základních dat o robotech, výchozí modul pro další rozšíření
- Maintenance Manager – plánování údržby a identifikace požadavků na údržbu
- Condition monitoring – sledování stavu robotu, sledování trendů, analýza mechanických a řídicích dat
- Messages – hlášení kritických událostí
- Notifications – zasílání informací o robotu pomocí emailu, definice události pro zaslání upozornění

Do budoucna KUKA slibuje modul pro prediktivní údržbu, spotřebu energie, dobu cyklu a další žádané funkce. Více informací o funkčnosti algoritmů, analyzovaných veličinách nebo konkrétnějším fungování systému iiQoT, není ve veřejně dostupných informacích uvedeno.

2 Analýza možností sběru dat

Naměření a sběr provozních dat stroje je klíčovým komponentem a prvním krokem k funkčnosti systému monitorování stavu robotu. Pro funkční analýzu stavu stroje je nutný neustálý přísun nových dat a s tím spojené neustálé monitorování stroje. Při monitorování je nutné získávat hodnoty v dostatečně častých intervalech, aby nedošlo ke ztrátě potřebné informace. Rychlé čtení a zaznamenávání hodnot přichází s větší náročností na hardware zařízení, na kterém k takové operaci dochází a při větším objemu dat je nutno myslet na zatížení způsobené tímto vlivem. Značnou zátěž na zařízení je možné způsobit i pouhým zasíláním dat za pomoci komunikačního rozhraní. Při komunikaci dat z PLC dochází k zatížení jeho procesoru a tím může dojít k omezení potřebného výkonu pro řídicí funkci PLC. Z těchto důvodů dochází při rozsáhlých monitorovacích systémech k implementaci samostatného hardwaru pro sběr, odesílání a případné zpracování dat. Systém je poté zcela nezávislý na řídicím systému a umožňuje využití jeho plné výkonnosti. Analýza možností sběru dat se zabývá možnostmi, které byly v rámci realizace práce uvažovány nebo použity.

2.1 Sběr dat

Data lze sbírat za pomoci velkého množství zařízení, senzorů a dalších prostředků. Pro účely návrhu systému monitorování lze zdroje dat rozdělit do několika základních kategorií.

2.1.1 Čtení dat z existujících zařízení

Získat data z již existujících zařízení na stroji patří mezi nejjednodušší možné metody. Zařízení v sobě obsahuje potřebná data a lze je využít pro další záznam a zpracování. Data v zařízeních se mohou týkat přímo zařízení a dochází k jejich výpočtu interně, příkladem může být jednotka robotického ramene, z které lze číst hodnoty provozních hodnot robotu. Zařízení mohou také shromažďovat data z připojených senzorů, jako tomu je u PLC, které má na své I/O připojeny senzory, které jsou prvotním zařízením pro získávání dat. Data jsou mezi zařízeními dále zasílána za pomoci komunikačních rozhraní.

2.1.2 Externí senzorka

V případě, že nelze potřebná data vyčíst z existujících zařízení, je nutné doplnit stroj o dodatečné senzorké vybavení pro záznam potřebných veličin. Senzor je umístěn na potřebné místo a dochází k napojení na zařízení pro sběr dat. Zde existuje množství řešení, jak toto napojení provést. Jednou možností je napojení nově přidaného senzoru na řídicí PLC stroje a přistupovat k datům přes PLC. Další možností je vyvedení senzoru do externí jednotky pro sběr dat. Tyto jednotky jsou vybaveny I/O pro připojení senzorů a nabízejí také možnosti pro sběr a záznam dat. Jednotky se používají například u veličin, které je nutno zaznamenávat vysokorychlostně a řídicí PLC stroje není vybaveno vysokorychlostními kartami. Výhodou je samostatnost systému, ale nevýhodou je nutnost použití dodatečného hardwaru.

2.2 Hardware pro sběr, záznam a vyhodnocení dat

Hardware pro sběr dat realizuje fyzické připojení jednotlivých snímačů, signálů anebo komunikační rozhraní pro přenos dat mezi zařízeními. Je vybaven komponenty pro příjem dat, připojení snímačů, zpracování signálů, komunikaci s jinými zařízeními a dalšími komponenty. Typů hardwaru pro sběr dat je velké množství a mohou se lišit v mnoha aspektech. Sběr dat může být realizován přes několik zařízení až do koncového zařízení, v kterém dochází k zpracování dat a jejich archivaci.

Řídicí systém (PLC)

Pokud je stroj řízen za pomoci PLC, je možné pro sběr dat využít přímo hardware PLC. Velkou výhodou je skutečnost, že PLC obsahuje velké množství dat o stroji, které používá pro jeho řízení. Jedná se o signály z jeho fyzických vstupů, nebo signály získané přes komunikační rozhraní s dalšími zařízeními. Příkladem můžou být roboty nebo pohony. Z existující struktury programu lze data jednoduše použít pro vlastní činnosti nebo ukládání dat. Nevýhodou této možnosti pro sběr dat je zatížení procesoru PLC příjmem, zpracováním a odesláním dat. V rámci této kapitoly diplomové práce je uveden pouze hardware, o kterém bylo při návrhu uvažováno nebo je na pracovišti použit.

iba hardware

Společnost iba AG se zabývá celým procesem záznamu procesního signálu. Nabízí řešení pro získání, záznam, zpracování, analýzu a vizualizaci provozních veličin stroje. Je ideálním nástrojem pro komplexní práci se signály stroje a nabízí širokou softwarovou výbavu pro jednotlivé kroky zpracování, ale také hardware pro čtení signálu. V portfoliu nabízí několik sériových řad zařízení sloužících pro získání, záznam a v některých případech také zpracování signálu. Každá řada má své specifické využití a nabízí jiné možnosti odkud bude zaznamenávaný signál získán.

iba - Field and Drive Buses

Zařízení nabízí záznam dat přímo z komunikačních sběrnic, na kterých je stroj provozován. Zařízení dokáže nezávisle na ostatních zařízeních odposlouchávat datový tok mezi řídicí jednotkou a připojenými periferiemi. Pro použití není nutno žádná speciální komunikační sběrnice. Zařízení je vyráběno v různých verzích podle sběrnice, na které bude použito. Podporovány jsou průmyslové sběrnice CAN, Profibus, PROFINET, EtherCAT, ABB, Ethernet/IP [16]. Pro záznam dat je použit integrovaný software ibaPDA, který je používán napříč všemi platformami a nabízí pokročilé funkce pro záznam a diagnostiku signálu.

iba - ibaMAQS

Jedná se o modulární systém zařízení na záznam a zpracování signálů [16]. Systém nabízí základní jednotku, která obsahuje základní možnosti pro záznam signálu, kterou je však možné rozšířit o další rozšiřující karty. Rozšiřující karty nabízejí další možnosti získání signálů a jedná se především o analogové nebo optické vstupy. Zařízení je schopno zaznamenávat data z velkého množství komunikačních rozhraní a také z dříve zmíněných vlastních vstupů. Mezi podporované zdroje dat patří PLC, sběrnice, MES systémy a další. Komunikace probíhá přes integrovaný ethernetový konektor. Data jsou zaznamenávána za pomoci softwaru ibaPDA a to přímo v zařízení. Zařízení je vybaveno také softwarem pro zpracování a analýzu dat ibaAnalyzer, který nabízí pokročilé možnosti práce se signály. Výstupem ze zařízení je soubor naměřených/zpracovaných dat, případně výstup do databáze. Zařízení tedy nabízí veškeré potřebné nástroje pro kompletní sledování stavu stroje a zpracování dat. Díky využití vlastního výpočetního výkonu dochází k minimálnímu ovlivnění okolních zařízení.

iba - The iba Modular system

Iba modular systém je starší ekvivalent ibaMAQS a nabízí také modulární provedení [16]. Hlavní jednotka obsahuje ibaPDA pro záznam signálů, ethernetový konektor pro propojení s jinými zařízeními a určité IO konektory. Rozšiřující karty nabízejí rozšíření IO konektorů. Zařízení je určeno především pro záznam z vlastních IO a nenabízí záznam signálů z PLC, sběrnic apod.

Omron zařízení pro sledování stavu

Omron nabízí sérii zařízení určených pro monitoring stavu motorů v průmyslu. Monitoring je realizován za pomoci malého modulu, který je zapojen na přívodní napájení motoru při analýze napájení motoru, anebo poblíž motoru při realizaci vibrodiagnostiky či termografie. Aktuálně Omron nabízí trojici modulů pro monitorování odebíraného proudu, vibrodiagnostiku a pro sledování izolačních odporů [17]. Jednotlivé moduly jsou soběstačné a nabízejí vlastní software pro nastavení parametrů analýzy. Moduly pracují s jistou formou KPI, která se mění v závislosti na změně stavu motoru. Jedná se však o číslo bez rozměru a jednotky, které se stanoví při prvním měření a poté se mění v závislosti na stavu sledovaného zařízení. Dále nabízí rozhraní pro předání informace o stavu motoru do nadřazeného systému.

V rámci ukázkové prezentace nám byl prezentován modul K6CM-CI2M, který provádí analýzu chodu motoru na základě jeho proudového odběru. Modul zachycuje poruchy okamžité a poruchy dlouhodobé. Za okamžitou poruchu může být považováno krátkodobé ovlivnění chodu motoru například zvětšením zátěže. Dlouhodobá analýza se zabývá plíživou chybou, která nastává v dlouhém časovém intervalu (roky) a projevuje se postupným zadíráním motoru.

2.3 Software pro sběr dat

Software pro sběr dat je nedílnou součástí řetězce pro zpracování dat. Softwary často umožňují vícero možností nejen pro sběr a záznam dat, ale také pro částečné vyhodnocení nebo pro další export dat. Softwarových nástrojů je na trhu, stejně jako hardwaru, velké množství. V této kapitole budou uvedeny pouze softwarové nástroje, které jsou v diplomové práci a na pracovišti využity.

Řídicí systém (PLC)

Jedním z možných softwarových nástrojů pro sběr, ukládání i analýzu dat je možnost přímého použití řídicího systému PLC. Do kódu PLC je možné za pomoci programovacích jazyků SCL, LAD a dalších implementovat logiku sloužící pro jednotlivé kroky zpracování dat. PLC nabízí širokou podporu logických i matematických funkcí, s kterými lze realizovat analýzy a zpracování dat. Obsahuje také knihovní prvky na komunikaci s databázovými systémy, které lze použít pro ukládání dat na databázový server. Použití řídicího systému PLC je na tyto úkony možné použít, oproti specializovaným programům zaměřených na práci s daty je však implementace složitější a přichází s dalšími nevýhodami. Největší nevýhodou je nutnost běhu zpracování dat přímo na řídicím systému, což ale může být u některých případů také jeho výhodou. Na jednu stranu není třeba dalšího zařízení, ale na stranu druhou je spotřebováván výkon PLC na práci s daty. Při malém rozsahu analýzy nebude vliv na výkon PLC tak razantní, ale při zpracování velkého množství dat může dojít k značnému ovlivnění výkonnosti PLC. Se zpracováním dat souvisí také složitější definice algoritmů za pomoci kódu v PLC, oproti specializovaným softwarům nabízející přizpůsobené prostředí zadávání datových analýz. Výhodou je přímý přístup k datům bez nutnosti dalších komunikací pro přenos dat.

Externí systém (iba)

Iba AG je německá společnost zabývající se komplexním řetězcem pro zpracování dat, od naměření až po vizualizaci analyzovaných dat. Na každou z oblastí práce se signálem nabízí vlastní aplikaci, která je přizpůsobená pro konkrétní úlohu a nabízí širokou řadu možností. Za „hlavní“ aplikaci lze považovat software ibaPDA, který slouží pro sběr a záznam signálů a dále poskytuje data dalším aplikacím v řetězci zpracování dat. ibaPDA nabízí širokou řadu rozhraní a produktů, z kterých je schopen odposlouchávat a zaznamenávat signály.

Mezi ty nejdůležitější komunikace a zařízení patří [18]:

Záznam dat z PLC systémů:

- PLC-Xplorer – Siemens, ABB, Beckhof, B&R, Mitsubishi, Rockwell, CODESYS, OMRON

Záznam dat z komunikačních rozhraní:

- Modbus (také varianta TCP)
- Profinet
- OPC-UA
- S7
- Raw-Ethernet
- EtherNet/IP
- Generic-TCP
- Generic-UDP
- MQTT

Záznam dat z databázových systémů:

- MySQL
- Oracle

Záznam dat z pohonů/měníčů:

- ABB Drives
- SIMOTION
- SINAMICS

Rozhraní, ze kterých je ibaPDA schopna zaznamenávat data je daleko více a nabízí velice širokou podporu skrze výrobce, typů zařízení a komunikačních protokolů. Software je schopen simultánního záznamu signálů z více zdrojů zároveň, a to v časech vzorkování od 1 ms. Rychlost vzorkování (záznamu dat) je samozřejmě závislé také na rychlosti komunikace a výkonu zařízení podílejících se na přenosu dat. Počet zaznamenávaných dat není nikterak omezován, co se týče schopnosti SW a závisí pouze na licenčním limitu pro počet signálů. Jednotlivá komunikační rozhraní jsou také často licencována samostatně. Záznam dat je realizován na základě podmínek pro start, stop nebo další nastavení. Zaznamenané průběhy signálů jsou ve výchozím nastavení ukládány do .dat binárního souboru. Ten je později možné využít programem ibaAnalyzer jako zdroj dat pro provedení zpracování. ibaPDA umožňuje export uložených dat také do dalších souborových formátů (.csv, .txt), na ibaHD-Server, do cloudu nebo do databáze. Jedná se o software pro platformu Windows, jak je ale zmíněno v 2. kapitole, je také implementován do specializovaného iba hardwaru. To umožňuje zařízení přímý záznam ze vstupních pinů iba zařízení. Některý iba hardware nabízí také integrovaný software pro zpracování dat ibaAnalyzer, jedná se tedy o komplexní řešení zpracování dat a to ve vlastním hardwaru.

2.4 Přenos dat

Pro přenosy dat mezi zařízeními jsou využívány vstupy, výstupy, rozhraní nebo komunikační protokoly. Následující výčet popisuje varianty použité v diplomové práci pro přenos dat.

PLC I/O

Jednou z nejjednodušších metod pro přenos informace je přivedení signálu za pomoci vodičů na vstupy, případně výstupy PLC. Signálové vstupy/výstupy mohou být integrovány do základního modulu PLC, nebo k PLC připojeny jako rozšiřující karty. I/O lze rozdělit do dvou základních kategorií, a to na digitální a analogové. Digitální signály nabývají pouze dvou hodnot (1 nebo 0), kdy každá úroveň odpovídá určitému napětí na I/O.

Používané rozsahy digitálních I/O:

- 0/24 VDC – nepoužívanější rozsah
- 0/5 VDC – TLL logika, používání zřídka
- 230 VAC – sepnutí střídavé silové sítě

Analogové I/O lze dělit do dvou kategorií dle jejich charakteru, a to na napěťové a proudové.

Používané rozsahy analogových I/O:

- 0 až 10 V – napěťový rozsah
- -10 až 10 V – obdoba, s využitím záporné polarity
- 0 až 20 mA – proudový rozsah
- -20 až 20 mA – obdoba, s využitím záporné polarity
- 4 až 20 mA – proudový rozsah s možností detekce přerušení vodiče

I/O slouží pro připojení jednotlivých snímačů, senzorů, relé, stykačů, indikátorů, tlačítek, ovládacích prvků a nespočtu dalších komponentů.

S7 Komunikace

S7 komunikace je vyvinuta a používána společností Siemens v jejich produktech od roku 1994, kdy došlo k představení produktové řady S7. Jedná se o komunikaci využívanou hlavně mezi zařízeními PLC, HMI a SCADA. Komunikace nabízí rychlé a jednoduché spojení mezi zařízeními na stejné síti. Datové spojení S7 komunikace je tvořeno datovou komunikací na TCP/IP protokolu. Díky tomuto standardizovanému protokolu je možné vytvořit vlastní HW nebo SW řešení podporující a integrující S7 komunikaci do vlastního systému. Takové integrace je použito u nástroje ibaPDA, umožňující navázání spojení s PLC a přístupem do datových bloků PLC.

Tabulka 2 – Struktura S7 komunikace [19]

Vrstva	ISO OSI vrstva	Protokol
7	Aplikační vrstva	S7 komunikace
6	Prezenční vrstva	S7 komunikace
5	Relační vrstva	S7 komunikace
4	Transportní vrstva	ISO-on-TCP (RFC 1006)
3	Síťová vrstva	IP
2	Linková vrstva	Ethernet
1	Fyzická vrstva	Ethernet

Díky TPC/IP protokolu je možné pro propojení zařízení využít standardní kroucené dvoulinky ethernet s konektory RJ45. Komunikace S7 telegramu se skládá z hlavičky, parametrů komunikace, parametrů dat a samotného datového bloku [19].

Pro komunikaci mezi PLC slouží interní funkce PUT a GET, které zasílají nebo čtou data z protější strany komunikace. Součástí programovacího prostředí jsou další knihovní prvky umožňující S7 komunikaci s dalšími zařízeními.

OPC UA

OPC UA (dále jen OPC) je standardizované komunikační rozhraní kontrolováno OPC Foudation. Nabízí rychlou, bezpečnou a spolehlivou komunikaci mezi zařízeními. OPC používá strukturu server-klient, kdy server poskytuje data a klient se k serveru připojuje pro jejich čtení. Jedná se o obousměrnou komunikaci a klient může data na server také nahrát. Komunikace OPC je postavena na obecně známých komunikačních standardech jako jsou TCP/IP a HTTP. Komunikace OPC definuje vlastní transportní vrstvu, komunikační vrstvu a aplikační vrstvu na ISO/OSI modelu [20].

Adresní prostor OPC komunikace je tvořen z uzlů (nodes), které mezi sebou mají vazby. Každý z uzlů disponuje vlastním identifikátorem (NodeId), který je složen z jmenného prostoru a unikátní části. Jmenná část neboli URI označuje identifikátor konkrétního uzlu a jedná se o unikátní označení. Uzly se v OPC komunikaci dělí na 8 základních tříd:

- Proměnné
- Typy proměnných
- Objekty
- Typy objektů
- Typy referencí
- Typy dat
- Metody
- Pohledy

Nejčastější třídou uzlu v zařízeních jsou proměnné, které obsahují samotnou hodnotu proměnné. Součástí je objekt vlastností, který může obsahovat upřesňující informace o dané proměnné. Komunikace mezi serverem a klientem probíhá za pomoci definovaných služeb (services), které slouží pro realizaci konkrétních funkcí OPC serveru. Pro vypsání dostupných proměnných je na serveru dostupná průzkumová služba, která vrátí seznam všech dostupných uzlů na serveru. Pro přenos dat poté slouží služby procházení, dotazování, metod, monitorování a odběru. Součástí komunikace je možnost zabezpečení za pomoci autentizace a certifikátů.

Modbus

Modbus je protokol pro vzájemnou komunikaci mezi zařízeními. Protokol dokáže fungovat na různých typech přenosových sítích a sběrnících. Mezi typická přenosová média patří Ethernet s TCP/IP, nebo sériová linka RS232/422/485, případně optické vlákno. Komunikace je založena na posílání zpráv mezi master a slave zařízeními. Master zařízení odesílá dotazy, na které slave zařízení v síti reagují a odesílají odpovědi. Master zařízení je vždy pouze jedno (např. PLC) a slave zařízení (senzor, měřicí karta, ale

také další PLC nebo robot) může být více. Modbus struktura je složena z přesně definovaných datových prvků, které jsou zobrazeny v následující tabulce.

Tabulka 3 – Datové typy Modbus struktury

Datový typ	Popis
Discrete Input	Binární vstup, pouze ke čtení
Coil	Cívka, umožňuje čtení i zápis
Input Register	16bit registr, pouze ke čtení
Holding Register	16bit registr, čtení i zápis

Tyto datové typy jsou dostupné v každém slave zařízení, ale může dojít k jejich úpravě na základě definice výrobce. Pro zápis nebo čtení jednotlivých proměnných je nutné z master zařízení poslat do slave zařízení příkaz pro daný úkon. Definování požadované funkce je realizováno za pomoci kódového označení příkazu, který se má v dané komunikaci použít.

Tabulka 4 – Kódové označení funkcí Modbus komunikace

Kódové označení	Funkce	Popis
01	Read Coils	Čtení bitů z Coils
02	Read Discrete Inputs	Čtení bitů z Discrete Inputs
03	Read Holding Registers	Čtení 16bit registrů z HR
04	Read Input Registers	Čtení 16bit registrů z IR
05	Write Single Coil	Zápis bitu na Coil
06	Write Single Register	Zápis 16bit registru
15	Write Multiple Coils	Zápis více bitů na Coil
16	Write Multiple Registers	Zápis více 16bit registrů

Při komunikaci dojde k definici funkce a adresního prostoru, který bude čten/zapisován. Na základě odeslaného příkazu dojde k obdržení čtených hodnot nebo zápisu do definované datové oblasti.

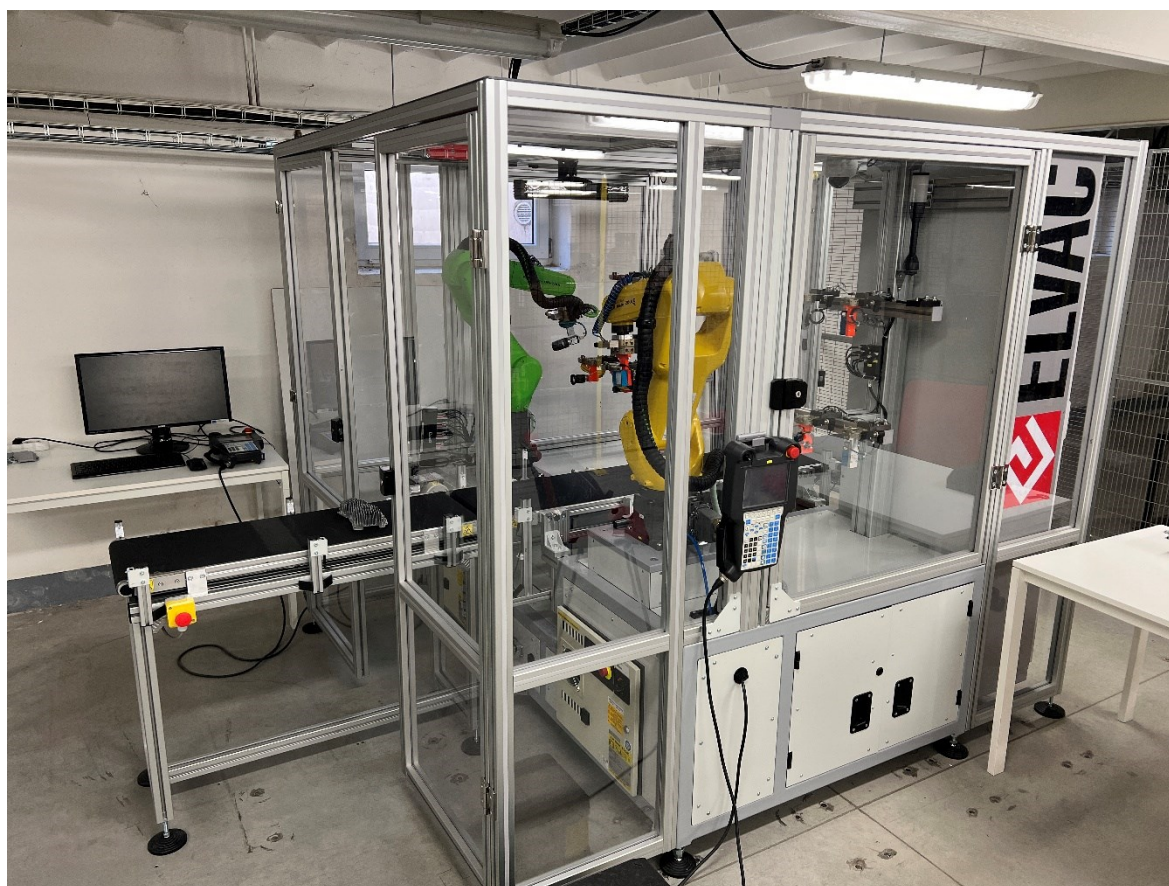
Další průmyslové komunikace

V případě robotických manipulátorů by bylo možné použití dalších průmyslových komunikací, pomocí kterých by bylo možné komunikovat s PLC a jinými systémy. Výbava pracoviště, na kterém je tato diplomová práce realizována však nemá tyto hardwarové nebo softwarové doplňky dostupné. Jedná se o možnost komunikací za pomoci Ethernet/IP, ProfiNET, ProfiBus, MQTT a další.

3 Popis robotického pracoviště

Praktická část diplomové práce je realizována na pracovišti TQC testeru, které slouží pro automatizované snímání výrobku za pomoci série kamerových kontrol výrobku. Pracoviště je vybaveno řadou výměnných kamer a osvětlovačů, které slouží jako výměnné nástroje pro dvojici robotických manipulátorů Fanuc. Napříč celým pracovištěm je veden pásový dopravník, po kterém se bude produkt pohybovat a na kterém bude docházet k jeho kontrole.

Proces kontroly výrobku probíhá následovně: Na páse je vložen analyzovaný objekt a za pomoci senzoriky dojde k automatickému spuštění pásu. Pás doveze výrobek na pozici mezi robotické manipulátory a zastaví se. Za pomoci 3D skeneru Photoneo dojde k pořízení 3D snímku objektu a definice požadovaných bodů inspekce. Inspekční body jsou pozice, ve kterých budou roboty realizovat kamerovou kontrolu. Zelený robot si odebere jednu z možných kamer a najede na první bod inspekce. Poté ho bude následovat žlutý robot, který s vybraným osvětlovacím přípravkem najede do blízkosti kamery a osvětlí snímanou část. Postupně se roboty provede inspekce na všech zvolených bodech a následně se inspekce opakuje s odlišnými kamerami a osvětlovači. Po dokončení všech inspekcí dojde k uvedení robotů zpět do výchozích pozic a pomoci pásu je výrobek dopraven na výstupní dopravník. Účelem stroje je realizace sady fotografií s různými konfiguracemi kamer a přisvětlovacích světel, ze které bude možné vybrat nejlepší možnou kombinaci pro snímání konkrétního prvku na sledovaném objektu (hrany, tvary, text apod.).



Obrázek 3 – Robotické pracoviště TQC testeru

3.1 Řídicí systém

Pracoviště je řízeno za pomoci PLC Siemens ET 200SP, které je vybaveno CPU 1510SP-1PN. PLC je rozšířeno o řadu digitálních, analogových a safety karet, které jsou využívány pro řízení pracoviště. PLC je hlavním řídicím členem pracoviště a slouží pro spuštění a řízení celého procesu analýzy výrobku. Během procesu komunikuje s roboty, kterým spouští robotické programy, nebo s kamerami ze kterých dostává informace o výsledku inspekce výrobku. Konkrétní označení PLC a všech přídatných rozšiřujících karet není pro účely této práce podstatné uvádět, jelikož použité funkce lze provozovat na kterémkoliv jiném PLC zařízení. Jedinou důležitou funkcí PLC, která je v práci využita, je funkce OPC UA klienta, kterou musí PLC podporovat a výbava analogovými vstupy pro fyzické měření proudu.



Obrázek 4 – PLC Siemens ET 200SP

Z hlediska této práce je důležité zmínit přítomnost dvou typů analogových karet, které je možné použít pro realizaci měření proudů robotu. Jedna z AI karet je základní s označením ST a druhá high speed s označením HS. V práci bude otestována použitelnost obou karet. Parametry karet jsou uvedeny v následující tabulce (Tabulka 5). Jeden z nejdůležitějších porovnávaných parametrů je cycle time karty, neboli čas nutný pro přečtení hodnoty na vstupech karty. Schopnost karty poskytovat co největší vzorkovací frekvenci hraje velkou roli při detekci špičkových proudů při pohybech robotu. Vlastnosti analogových karet ve spojitosti s měřeními průběhy budou porovnány a testovány v následujících kapitolách práce.

Tabulka 5 – Parametry analogových PLC karet [21], [22]

Parametr	Karta AI 4x U/I 2-wire ST	Karta AI 2x U/I 2-/4-wire HS
SN	6ES7134-6HB00-0DA1	6ES7134-6HD01-0BA1
Počet kanálů	4	2
Vstupní rozsahy napětí, rozlišení	0–10 V, 15bit 1–5 V, 15bit -10–10 V, 16bit -5–5 V, 16bit	0–10 V, 15bit 1–5 V, 13bit -10–10 V, 16bit -5–5 V, 15bit
Vstupní rozsahy proudu, rozlišení	0–20 mA, 15bit 4–20 mA, 15bit	0–20 mA, 15bit -20–20 mA, 16bit 4–20 mA, 14bit
Převodník	Integrační, sigma-delta	Postupná aproximace
Čas převodu (na každý kanál)	180 / 60 / 50 ms	10 μ s
Doba cyklu (všechny kanály)	Součet časů převodů všech aktivních kanálů	125 μ s

3.2 Robotické manipulátory

Pracoviště je vybaveno robotem Fanuc LR Mate 200iD a kolaborativním robotem Fanuc CR – 7iA/L. V obou případech se jedná o 6 osé roboty s kompaktními rozměry a nabízejí dosah přibližně 70 cm. Maximální nosnost robotů je 7 kg. Roboty, dle popisu pracoviště, slouží pro kamerové kontroly analyzovaných komponentů. Jeden robot slouží pro pozicování kamery a druhý na pozicování přípravku s přisvitem. Oba roboty jsou vybaveny šesticí výměnných přípravků s různými typy nástrojů.



Obrázek 5 – Fanuc LR Mate 200iD



Obrázek 6 – Fanuc CR – 7iA/L

Oba roboty jsou vybaveny řídicí jednotkou R-30iB, která slouží pro řízení robotu. S jednotkou je robot spojen za pomoci jednoho kabelu s harting konektory. Jednotka robotu je připojena do sítě pracoviště za pomoci ethernetového kabelu a umožňuje navázání komunikace s dalšími systémy.

Tabulka 6 – Parametry robotů

Robot	Počet os	Dosah (cm)	Nosnost (kg)	Řídicí jednotka	Rok výroby
Fanuc LR Mate 200iD	6	70	7	R-30iB Mate Plus	2021
Fanuc CR – 7iA/L	6	70	7	R-30iB Mate Plus	2018

Řídicí jednotky robotů jsou dále vybaveny dotykovým panelem iPendant, který slouží pro parametrizaci, programování a ovládání robotu. Je složen z dotykového displeje a kláves pro ovládání prostředí pendentu. Součástí pendentu je tlačítko nouzového zastavení, které robota okamžitě zastaví.



Obrázek 7 – Fanuc iPendant

3.2.1 Motorová výbava robotů

Jednotlivé osy robotu jsou ovládány za pomoci servomotorů, které zajišťují jejich pohyb. Typy použitých motorů jsou uvedeny v dokumentaci robotu a také na štítku umístěném na těle robotu. U robotu jsou použity třífázové střídavé servomotory odlišného typu pro konkrétní osu.

Tabulka 7 – Typové označení a parametry motorů robotu Fanuc LR Mate 200iD

Motor	Typ	Fáze	Model	Napětí (V)	Frekvence (Hz)
M1	AC Servo	3f	BISR1/6000	172	400
M2	AC Servo	3f	BISR1/6000	172	400
M3	AC Servo	3f	BISR0.5/6000	86	400
M4	AC Servo	3f	BISR0.4/6000	31	267
M5	AC Servo	3f	BISR0.2/6000	50	400
M6	AC Servo	3f	BISR0.2/6000	50	400

Tabulka 8 – Typové označení a parametry motorů robotu Fanuc CR – 7iA/L

Motor	Typ	Fáze	Model	Napětí (V)	Frekvence (Hz)
M1	AC Servo	3f	BISR1/6000	172	400
M2	AC Servo	3f	BISR1/6000	172	400
M3	AC Servo	3f	BISRO.5/6000	86	400
M4	AC Servo	3f	BISRO.4/6000	31	267
M5	AC Servo	3f	BISRO.2/6000	50	400
M6	AC Servo	3f	BISRO.2/6000	50	400

3.2.2 Předepsaná údržba robotů

U robotů je servisní příručkou [23] přesně předepsaná údržba, která by měla být vykonávána pro udržení optimální výkonnosti robotu a zamezit jeho poruchám. Informace o údržbě jsou uvedeny v manuálu pro operátory a obsahuje podrobné informace, na kterých částech a jak často by měla být údržba prováděna. Údržbu je nutné dle manuálu provádět v přesně daných intervalech, a to konkrétně jednou ročně za předpokladu, že dochází k provozu robotu do 3840 hodin ročně. V případě provozu robotu více než 3840 hodin ročně je nutný úměrný přepočítání intervalu pro častější kontroly.

Denní kontrola

Jedná se o seznam kontrol doporučených výrobcem, které je vhodné provádět před každodenním provozem. Jedná se o ruční čištění komponentů a o vizuální kontrolu stavu robotu.

- Únik oleje – kontrola olejových stop okolo jednotlivých os robotu. Olej se může akumulovat na krytech os robotu v závislosti na pohybu a prostředí. V případě výskytu oleje je nutné jeho otření, aby došlo k zamezení vzniku nahromadění oleje a vytvoření olejové skvrny pod osou. Průsaky mohou být způsobeny provozem ve vysokých teplotách a tím umožnit vznik většího tlaku uvnitř osy.
- Vzduchová soustava – kontrola tlaku vzduchu v soustavě.
- Vibrace, hluk – kontrola, zda nedochází k neobvyklým vibracím robotu nebo k vydávání nestandardních zvuků při provozu.
- Opakovatelnost – kontrola, zda nedochází k zřetelné odchylce pozice robotu při najetí do známe pozice s ohledem na předchozí cykly.
- Periferie robotu – kontrola správné komunikace a funkčnosti připojených periférií robotu (gripper apod.).
- Brzdy os – kontrola funkce brzd os. Kontrola se provádí vypnutím robotu a sledováním, že nedojde k vychýlení efektoru robotu o více než 2 mm.
- Varování – kontrola chybových hlášek na operátorském panelu.

Pravidelná kontrola a údržba

Pravidelné prohlídky robotu jsou definovány v daných časových intervalech, které vycházejí z hodin provozu robotického ramene. Pro provoz robotu 3840 hodin je definovaný cyklus kontroly jednou ročně. S navyšujícím počtem hodin provozu robotu dochází k úměrnému zkracování doby pravidelné kontroly. Kontroly prováděné v ročním intervalu obsahují výjimku, kdy první kontrolu je nutné provést již 3 měsíce (960 hodin provozu) po uvedení robotu do provozu. Po této prvotní kontrole již platí provádění kontroly v ročních intervalech.

Kontrola a údržba prováděna v 3měsíčním cyklu (960 hodin provozu):

- Čištění ventilačního systému řídicí jednotky robotu
- Kontrola vnějšího poškození a odlupující se barvy
- Kontrola vystavení působení kapalin (voda, olej)

Kontrola a údržba prováděny v ročním cyklu (3840 hodin provozu):

- Kontrola stavu kabelů k řídicí jednotce, robotu a operátorskému panelu
- Kontrola stavu efektoru, kabelů k efektoru a kabelů k externím bateriím
- Kontrola nezakrytých konektorů
- Dotažení šroubů na efektoru
- Dotažení šroubů upevňující robota
- Kontrola mechanické zarážky osy J1/J3
- Čištění prachu, pilin či zbytků barvy z těla robotu
- Výměna baterií robotu

Kontrola a údržba prováděna v 4letém cyklu (15360 hodin provozu):

- Doplnění maziva do jednotlivých os robotu
- Výměna kabelu mezi robotem a řídicí jednotkou
- Výměna baterií v řídicí jednotce

Pravidelná výměna baterií

Robot je vybaven záložními bateriemi, které zajišťují uchování informací o pozicích jednotlivých os robotu a jejich pravidelnou výměnou je předcházeno ztrátě uchovávaných informací. V případě ztráty aktuálních pozic je nutné opětovné provedení kalibrace robotu. Baterie jsou umístěny pod krytem v základně robotu. Při použití vestavěných baterií je nutná výměna jednou ročně a při použití externích baterií jednou za rok a půl. Výměna je taky nutná při poklesu napětí na bateriích, který je signalizovaný za pomoci alarmu v ovládacím panelu robotu. Výměnu je nutné provádět při zapnutém napájení, aby nedošlo ke ztrátě informací po dobu výměny baterií a při aktivovaném nouzovém zastavení pro zamezení pohybu robotu.

3.2.3 Funkčnost robotu bez záložních baterií

V průběhu práce s roboty došlo k vypršení intervalu pro výměnu baterií a bylo nutné tento servisní zákrok řešit. O nutnosti výměně baterií informovaly oba roboty za pomoci alarmů na ovládacích panelech. Alarmy signalizovaly pokles napětí na bateriích do takové úrovně, kdy už nebyly schopny napájení elektroniky pro uchování informací o pozicích robotu. V takovém případě dochází po vypnutí napájení robotu ke kompletní ztrátě veškerých informací o poslední pozici robotu a pozic enkodérů na jeho osách. Provoz robotu bez záložních baterií, a to i při neustálém připojení do sítě, je však téměř nemožný. Pokud dojde k aktivaci alarmu signalizující nutnost vyměnit baterie a již došlo ke ztrátě pozic robotu, dojde k omezení veškerých pohybových funkcí robotu. Robot v takovém stavu neumožňuje spouštění programů, přejetí na pozici, a dokonce ani JOG jednotlivých os. Současně robot požaduje kalibraci os. Uvedení robotu zpět do provozuschopného stavu je možné pouze za pomoci výměny vadných baterií, potvrzení výměny a restartu robotu. Po provedení procedury výměny je nutná opětovná kalibrace robotu. Po těchto krocích dojde k opětovnému povolení pohybových funkcí robotu a možnosti normálního fungování.

3.2.4 Dostupné komunikační rozhraní robotu

Roboty Fanuc nabízejí několik komunikačních rozhraní, pomocí kterých dokáží komunikovat s dalšími zařízeními. Většinu komunikačních rozhraní je nutné zakoupit jakožto extra softwarové, nebo hardwarové vybavení robotu. Mezi nabízené komunikační protokoly Fanuc robotů patří:

- Modbus TCP – vyžaduje zakoupení SW licence
- Ethernet/IP – vyžaduje zakoupení SW licence
- Profinet – vyžaduje zakoupení rozšiřující HW karty
- EtherCat – vyžaduje zakoupení rozšiřující HW karty
- DeviceNet – vyžaduje zakoupení SW licence
- CC-Link – vyžaduje zakoupení rozšiřující HW karty

Na využívaném robotu Fanuc LR Mate 200iD je aktivní SW licence na Modbus TCP, pomocí kterého dochází k řízení robotu za pomoci PLC. Pro účely řízení je tato komunikace dostatečná, ale pro účely sledování stavu a rychlého vyčítání provozních hodnot robotu je rychlost nedostačující. Robot je dále vybaven OPC UA serverem, který nabízí komunikaci předem vybraných proměnných robotu, ale také uživatelsky zvolených proměnných v Modbus struktuře. Rychlost komunikace za pomoci OPC UA je násobně rychlejší nežli přes Modbus. Dokumentace uvádí minimální čas 100 ms pro čtení proměnné robotu.

3.2.5 Vnitřní proměnné robotu

Robot nabízí řadu typů proměnných, které slouží pro programování robotu, ukládání pozic, komunikace s okolím, nastavení nebo další. Každý typ je v robotu využíván za jiným účelem a záleží na konkrétním použití.

Programové registry (interní proměnné)

- R[x] – registry
- PR[x] – poziční registry
- F[x] – Flags
- G[x] – Groupy (skupiny)

Komunikační proměnné (dostupné pro Modbus)

- DI[X] – digitální vstupy
- DO[X] – digitální výstupy
- \$snpx.asg[80] – přiřazení volitelných proměnných do holding registru (Modbus data model dostupný přes OPC UA)

Systémové proměnné

- UOP – přiřazení systémových proměnných pro řízení z jiného systému (PLC)
- SOP – přiřazení systémových proměnných při řízení z panelu Fanuc
- \$config – systémové nastavení robotu
- Variables – veškeré proměnné robotu

3.2.6 Dostupnost proměnných v OPC UA

Roboty Fanuc na pracovišti nabízejí integrovaný OPC UA server, který poskytuje data z robotu. Jedná se jak o předem vybrané proměnné, tak uživatelsky definované proměnné. Jejich odesílání probíhá přes Modbus strukturu a přiřazení do jednoho z možných polí.

OPC UA server nabízí nodes, neboli uzly (lze si představit jakožto složky), které obsahují jednotlivé proměnné. Jeden ze základních je node „RobotInformation“, který poskytuje základní data o robotu a jeho stavu. Obsahuje následující proměnné:

- Alarm – informace o alarmovém hlášení
- ModeState – mód robotu (Auto, T1, T2)
- Model – modelové označení
- OperationState – číslo stavu robotu
- Position – aktuální pozice robotu v prostoru, vrací pole X, Y, Z, W, P, R
- ProgramSpeed – aktuální nastavená rychlost v procentech
- SerialNumber – sériové číslo robotu
- ServoState – stav servopohonů (on/off)
- Torque – pole Q složek proudu jednotlivých os
- Uptime – čas napájení robotu
- Verison – označení SW verze kontroléru robotu

Další dostupný node v OPC UA komunikaci je Modbus node. Ten obsahuje standardizovanou Modbus strukturu, kterou lze za pomoci OPC komunikace číst nebo do ní zapisovat. Přístup k daným proměnným robotu je definován za pomoci adresy, na které se proměnná nachází, nebo lze přiřadit proměnné dle vlastního uvážení do holding registru. Možnost komunikace vlastních hodnot je testována a popsána v dalších kapitolách.

Tabulka 9 – Modbus struktura proměnných robotu definována manuálem [23]

Typ proměnné	Rozsah adres	Príslušná robotická data (a: adresa)
Discrete input	1 ~ 10000	Digital input DI[a]
	10001 ~ 20000	Robot input RI[a-10000]
	20001 ~ 21000	UOP input UI[a-20000]
	21001 ~ 21999	UOP output UO[a-21000]
	22000 ~ 22999	SOP input SI[a-22000]
	23000 ~ 24000	SOP output SO[a-23000]
	24001 ~ 25000	Weld interface digital input WI[a-24000]
	25001 ~ 26000	Weld interface digital output WO[a-25000]
	26001 ~ 27000	Wire soldering detector input WSI[a-26000]
	27001 ~ 28000	Wire soldering detector output WSO[a-27000]
	28001 ~ 65536	Nepoužito
Coils	1 ~ 10000	Digital output DO[a]
	10001 ~ 20000	Robot output RO[a-10000]
	20001 ~ 30000	Flag F[a-20000]
	30001 ~ 65536	Nepoužito
Input Registers	1 ~ 1000	Group input GI[a]
	1001 ~ 2000	Group output GO[a-1000]
	2001 ~ 3000	Analog input AI[a-2000]
	3001 ~ 4000	Analog output AO[a-3000]
	4001 ~ 65536	Nepoužito
Holding Registers	1 ~ 16384	Uživatelsky definovaná data (R[a] ve výchozím nastavení)
	16385 ~ 16448	55.4 dynamické přiřazení robotických dat do tohoto adresního prostoru
	16449 ~ 65536	Nepoužito

Tabulka 10 - Oprávnění pro čtení/zápis do Modbus struktury definována manuálem [23]

Typ proměnné	Datová reprezentace proměnné	Oprávnění
Discrete input	1 bit	Čtení
Coils	1 bit	Čtení, zápis
Input Registers	16-bit word	Čtení
Holding Registers	16-bit word	Čtení, zápis

Z každého nodu lze pomocí OPC UA komunikace vyčítat hodnoty, případně zapisovat, pokud to daný node umožňuje. Z tabulek (Tabulka 9, Tabulka 10) lze vidět, že každý ze základních nodes obsahuje datový prostor 65536 bytů, který je dále rozdělen na konkrétní datové rozsahy definující přístupy k datům robotu. Uživatelsky definované proměnné jsou ukládány do prostoru holding registru. V základní konfiguraci je do něho přiřazen vnitřní registr R[x-y], ale lze tento prostor namapovat na libovolné proměnné dostupné v robotu. A to dokonce také systémové proměnné, které jsou uloženy hluboko v nastavení robotu. Přiřazení proměnných probíhá pomocí ovládacího panelu robotu za pomoci přiřazení proměnných do systémové proměnné \$snpx.asg[80].

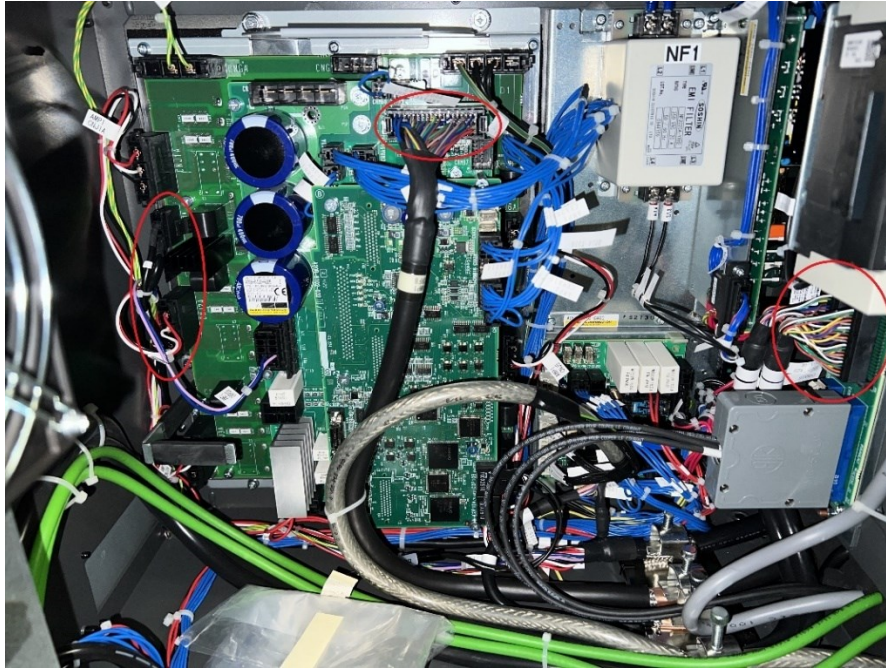
3.3 Komponenty externího měření proudů os robotu

Z důvodu omezených komunikačních rychlostí robotu, a také z důvodu vývoje systému, který by byl kompletně nezávislý na datech z řídicí jednotky robotu, došlo k realizaci testovacího měření přímo na přívodním kabelu robotu. Dochází tedy k měření proudů na jednotlivých osách robotu a umožňuje vlastní zaznamenávání průběhu proudu během pohybu robotu. Aby docházelo k měření proudů, které přímo pohánějí servomotory jednotlivých os, je nutné umístit měřicí zařízení na kabel mezi kontrolér robotu a samotný robot. Jedná se o jediný kabel, kterým je robot s kontrolérem spojen a poskytuje veškeré komunikační a silové vodiče.

Realizace měření obnášela řadu překážek, které bylo nutné pro realizaci měření vyřešit. Hlavní překážkou bylo, jakým způsobem bude měření realizované, aby nedošlo k poškození spojení mezi robotem a kontrolérem, případně poškození propojovacího kabelu. Dodávaný kabel obsahuje harting konektor (Obrázek 8) na straně robotu a na straně kontroléru je kabel rozčleněn do několika menších konektorů, které jsou přímo zapojeny do obvodů kontrolérů. Rozdělení do konektorů v rámci kontroléru je značeno červenými kruhy (viz Obrázek 9).

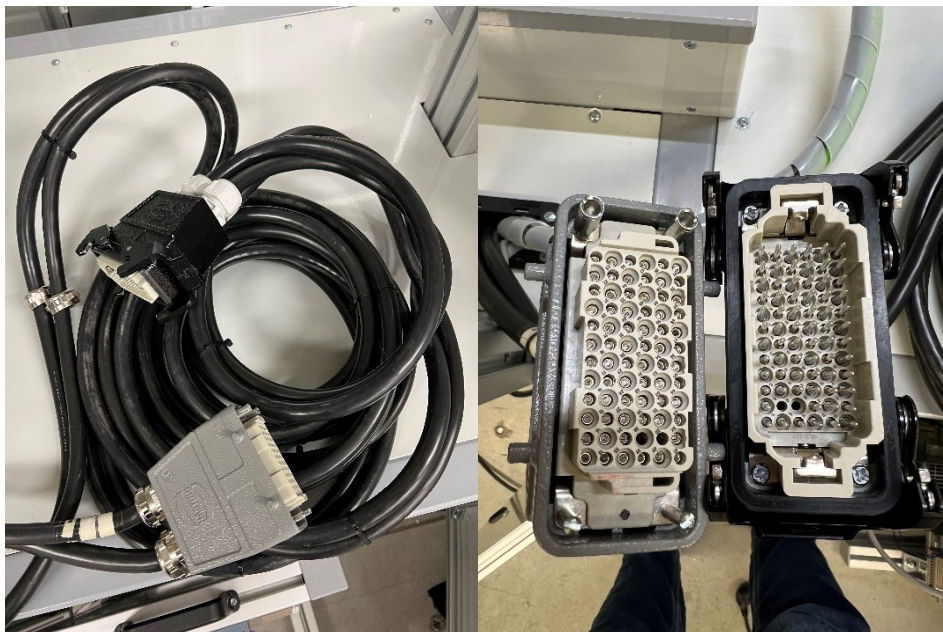


Obrázek 8 – Připojení kabelu do robotu – harting konektor



Obrázek 9 – Připojení kabelu do kontroléru – rozdělení na více svazků

Pro měření konkrétních žil napájecích vodičů servomotorů robotu, bylo nutné najít místo kde by bylo možné umístit měřicí kartu. Až na konektorech v řídicí jednotce robotu jsou žíly obaleny vrstvou izolace a stínění. Na kabelu robotu tedy není žádné místo, kde by bylo možné provést neinvazivní měření. Aby při realizaci měření nedošlo k poškození hlavního kabelu, byl po konzultaci s Fanuc podporou pořízený prodlužovací kabel. Kabel obsahoval harting konektory na obou stranách a byl zapojen mezi stávající kabel z kontroléru a robotu. Účelem pořízení tohoto kabelu byla možnost porušení jeho izolace, vytažení potřebných žil a realizace měření bez nutnosti poškození originálního kabelu k robotu, který byl součástí instalace robotu na pracovišti.



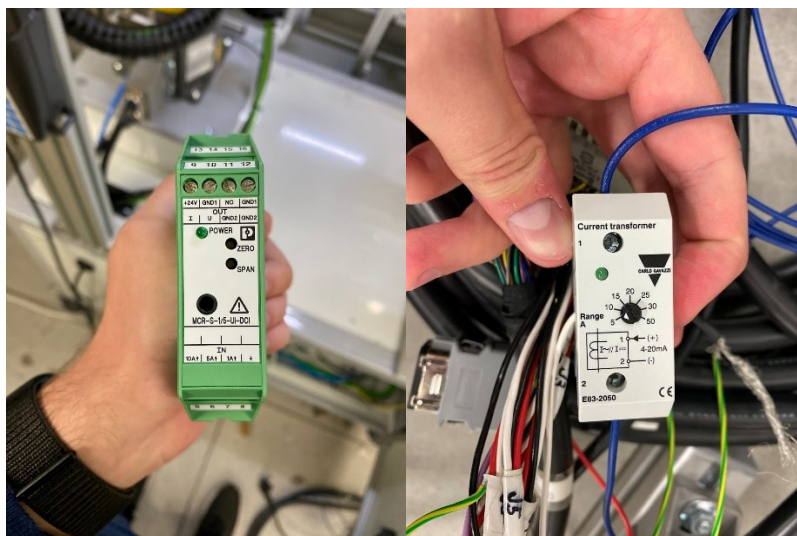
Obrázek 10 – Prodlužovací kabel k robotu

Pro měření byly zvoleny měřicí proudové transformátory, které měří proud procházející připojeným vodičem. Další důležitou částí byla identifikace, které vodiče slouží pro řízení motorů a na kterých lze proud měřit. Jak vyplývá z obrázku harting konektoru (Obrázek 10), robot je připojen za pomoci 72 vodičů a nelze přesně určit, které vodiče slouží k napájení motorů os. Informace o specifikaci vodičů nejsou veřejně dostupné, a to hlavně z důvodu možného obcházení Fanucu při realizaci vlastních kabelů. Pro účely diplomové práce mi byly tyto informace poskytnuty, ale nemohl jsem je dále šířit a zveřejnit. Každá osa je napájena trojfázově, tedy jeden vodič pro každou fázi. Při předpokladu vyrovnaného zatížení jednotlivých fází postačuje měření na jednom fázovém vodiči motoru.



Obrázek 11 – Ukázka rozloženého harting konektoru

Pro měření proudů na napájecích vodičích motorů robotu jsou použity měřicí transformátory proudu. Konkrétně byly specifikovány 2 měřicí trafo, které jsou v rámci práce dále porovnávány. Jedná se o měřicí proudové trafo Carlo E83-2050 a Phoenix contact MCR-S-1/5-UI-DCI. Trafo od výrobce Carlo je průvlastkové a je napájené samotnou měřicí smyčkou. Trafo od Phoenix contactu potřebuje pro svůj provoz vlastní napájení a zároveň je nutné přivést měřený vodič na požadované svorky měřicí karty. Z pohledu principu tedy karty pracují totožně, jejich provedení je ale rozdílné a v následujících kapitolách je uvedeno jejich porovnání z hlediska vhodného použití pro měření proudu na jednotlivých osách robotu.



Obrázek 12 – Měřicí transformátory proudu

Tabulka 11 – Parametry měřicích proudových transformátorů [24], [25]

Parametr	Carlo E83-2050	Phoenix contact MCR-S-1/5-UI-DCI
Připojení měřeného vodiče	Průvlakové	Připojení na svorky
Napájení	Napětovou měřicí smyčkou (10–40 VDC)	Externí napájení 20–30 V DC
Měřený proud	0–50 A	0–11 A
Měřený rozsah proudu	0–5 A, 0–10 A, 0–15 A, 0–20 A, 0–25 A, 0–30 A, 0–50 A	1 A vstup: 0–1 A, 0–0,65 A, 0–0,4 A, 0–0,25 A 5 A vstup: 0–5 A, 0–3,25 A, 0–2 A, 0–1,25 A 10 A vstup: 0–10 A, 0–6,5A
Výstupní signál	4–20 mA	Proudový: 0–20 mA, 4–20 mA Napětový: 0–10 V, 0–5 V, -10–10 V, -5–5 V A další varianty.
Tolerance výstupního signálu	$\pm 2 \% @ 50 \text{ Hz}$	< 0,5 % za jmenovitých podmínek
Čas odezvy	T < 200 ms	40 ms with DC 330 ms with AC
Indikační dioda o provozu	Ano, zelená	Ano, zelená
Provozní teplota	-20–50 °C	-20–60 °C
Nastavení rozsahů a parametrů	Otočný přepínač pro nastavení rozsahu měřeného proudu	Přepínače pro nastavení měřeného rozsahu, výstupu a dalších parametrů. Ladící potenciometry pro nastavení nuly a doladění měřeného rozsahu.

Pro měření proudů os robotu jsou využívány proudové měřicí transformátory, které jsou konstruovány pro měření velkých střídavých proudů. Typický proudový transformátor se skládá z feromagnetického toroidu, na kterém je navinuto n závitů měděného drátu, který se chová jako sekundární cívka transformátoru. Měřený vodič je umístěn do středu toroidu a působí jako primární cívka. Vinutí sekundární cívky je přivedeno na snímací odpor, který odpovídá požadované snímané sensitivitě nebo rozsahu. Při průchodu proudu v primárním vedení začne docházet k vzniku kruhového magnetického pole H. To způsobí, že se na sekundárním vinutí začne při průchodu elektrického pole indukovat napětí. Napětí na rezistoru způsobí vznik proudění proudu, pomocí kterého lze detekovat velikost proudu proudící primárním vodičem. Indukovaný proud je však násobně menší nežli měřený. [26] Tohoto principu je využíváno u obou měřicích karet.

4 Návrh systému vyhodnocení stavu pracoviště a sběru dat

Za účelem realizace návrhu systému monitorování robotu byly stanoveny sledované proměnné a algoritmy, pomocí kterých lze stav sledovat a které jsou vypovídající o změnách provozu robotu.

4.1 Analyzované veličiny

Pro realizaci sledování stavu robotu je nutno zvolit vhodné parametry a veličiny, které dokáží reprezentovat stav robotu. Jedná se o veličiny, které jsou závislé na stavu robotu a při výskytu poruchy by došlo k jejich změně. Jelikož se práce zaměřuje převážně na detekci možných opotřebení mechanických částí robotu, jsou voleny veličiny týkající se průběhů na motorech jednotlivých os robotu.

4.1.1 Proud

Každá z os robotu obsahuje vlastní elektrický servomotor, který slouží k pohybu osou. Modelové označení a parametry jednotlivých motorů jsou uvedeny v předchozí kapitole zabývající se vybavením pracoviště. Jednotlivé servomotory mají předem dané napětí na kterém jsou provozovány, veličina, která se ale v závislosti na zatížení a pohybu robotu mění, je odebíraný proud. Při větší zátěži robotu dochází k nutnosti vyvinutí většího výkonu a tím zvýšení odběru proudu. Tohoto konceptu je využito při celém konceptu condition monitoringu. Při opotřebení nějaké z části motoru, převodovky, ložisek, nebo jiných částí, dojde ke zvýšení odporu pohybu mechanických částí robotu a motor bude nucen tento odpor překonat za pomoci zvýšení potřebného výkonu pro svou činnost. Díky tomuto principu lze za pomoci sledování proudu detekovat změny na zatížení nebo mechanických částech robotu. Roboty Fanuc, s kterými v rámci práce pracuji, poskytují následující proměnné proudu:

Q_CURRENT##

- Q fázové proudy os robotu
- Jednotky Ampéry (A)
- Dostupné v rámci OPC-UA serveru (vysoká rychlost záznamu dat)

CURRENT##

- Proudů os robotu
- Jednotky Ampéry (A)
- Systémové proměnné, nutno posílat přes holding register (nízká rychlost záznamu dat)

4.1.2 Točivý moment (Torque)

Dle Fanuc dokumentace je jako Torque označován Q fázový proud jednotlivých os robotu. Tedy proud, který je potřeba k vytvoření točivého momentu osy. Jeho velikost je udávána v ampérech (A). Proměnné jsou umístěny hluboko uvnitř systémových proměnných robotu nebo na OPC UA serveru:

- \$MOR_GRP[X].\$CUR_DIS_TRQ[X]
- \$MOR_GRP[X].\$TORQUE[X]
- OPC UA / TORQUE – REÁLNĚ Q_CURRENT

4.1.3 Teplota motorů

V rámci proměnných robotu jsou dostupné také teploty jednotlivých servomotorů os robotu. Hodnoty teplot motorů jsou důležité pro indikaci přetížení motoru, nebo nedostatečného chlazení a pro realizaci systému monitoringu robotu jsou data o teplotách motoru sbírána a vyhodnocována. Fanuc uvádí

teplotu jednotlivých os v rozmezí od 0 – 100 % a nelze z dostupného údaje definovat konkrétní aktuální teplotu motoru osy ve °C. Známa není ani maximální teplota motoru. Z procent však stále lze definovat podmínku, kdy dochází ke kritickému zahřívání jedné z os a vzniku alarmu či jiného upozornění. Za teploty blížící se nebezpečí přehřátí motoru lze považovat hodnoty blížící se vrchní hranici intervalu.

Teploty motorů lze najít v systémových proměnných robotu a jsou uloženy v poli \$MOR_GRP[1].\$TEMPERATURE[1...9]. Hodnoty teplot motorů jsou sice na první pohled dostupné napřímo přes OPC UA komunikaci, ale při jejich čtení se vždy rovnají hodnotě NULL a ani při zahřátí robotu se nezmění. Je možné, že se jedná o špatné nastavení robotu nebo nějakého parametru, ale na robotu, na kterém je realizována tato práce byly vždy rovny hodnotě null. Došlo tedy k nutnosti vytvoření BG tasku (program běžící na pozadí), který hodnoty ze systémových proměnných četl a zapisoval na výstupy robotu. Z výstupů již byly hodnoty za pomoci existující Modbus komunikace přeneseny do PLC. Došlo tedy pouze k rozšíření komunikační struktury.

4.2 Algoritmy stavu pracoviště

Pro vyhodnocení stavu stroje je možné využít statistických metod hodnocení. Ty nabízejí základní, ale také pokročilé postupy, které lze pro monitorování využít. Výhodou je jednoduchost metod a jasný výstup daný jedním číslem charakterizující celý analyzovaný průběh. Tyto výstupy pak lze velmi jednoduše porovnávat v časové oblasti a detekovat změny mezi analyzovanými intervaly. Jednotlivé metody je možné navzájem kombinovat a dosáhnout tak komplexnější analýzy pozorovaného průběhu. Pro definice statistických metod bylo čerpáno z knihy zabývající se statistickými metodami zpracování dat [27], [28].

Maximální hodnota

Maximální hodnota je definována jako největší hodnota v analyzovaném intervalu hodnot. Výpočet je možné realizovat na základě porovnání všech hodnot v intervalu. Prostředí programů již často nabízejí integrovanou funkci max, kterou je pro výpočet možné použít. Maximální hodnota je důležitá při detekci špiček signálu, které by jiné metody nezaznamenaly. V případě analýzy proudů se jedná o jedno z nejdůležitějších statistických hodnocení, jelikož s narůstajícím výkonem motoru rostou špičkové proudy při startu motoru.

$$y = \max(M), \text{ kde } M \text{ náleží analyzovanému intervalu hodnot} \quad (1)$$

Aritmetický průměr

Aritmetický průměr je statistická veličina, která určuje průměrnou hodnotu z analyzovaného intervalu. V běžné řeči je často označován obecným pojmem průměr a jedná se o pravděpodobně nejčastěji používaný statistický pojem. Obvyklé značení aritmetického průměru je vodorovná čára nad proměnnou. Výpočet je prováděn jako součet všech prvků analyzovaného intervalu, který je podělen jejich počtem.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \text{ kde } n \text{ náleží počtu proměnných } x \quad (2)$$

Medián

Medián je hodnota, která dělí analyzovaný interval na dvě poloviny se stejným počtem hodnot na obou stranách. K výpočtu mediánu je nutné seřadit hodnoty podle velikosti a určit hodnotu nacházející se uprostřed. Při sudém počtu hodnot v intervalu je z pravidla prováděn aritmetický průměr z dvou

prostředních hodnot. Musí však platit, že polovina hodnot intervalu je menší nežli medián a druhá polovina větší. Výpočet je možné realizovat za pomoci vlastních postupů, ale většina prostředí již má pro medián implementovanou vlastní funkci.

$$\tilde{x} = \text{median}(M), \text{ kde } M \text{ náleží analyzovanému intervalu hodnot} \quad (3)$$

Koeficient šikmosti (Skewness)

Koeficient šikmosti je označován jako γ_1 , charakterizuje rozdělení náhodné veličiny a popisuje její nesymetrii. Šikmost může nabývat kladné, záporné, ale také nedefinované hodnoty. Hodnotu šikmosti ovlivňuje rozložení náhodné veličiny vzhledem ke střední hodnotě. V případě, že se vlevo od průměru nachází většina hodnot, nazýváme šikmost jako kladnou. V případě umístění většiny hodnot vpravo od průměru se jedná o šikmost zápornou. Pokud by rozdělení bylo symetrické, byla by šikmost rovna nule.

$$\gamma_1 = \frac{\mu_3}{\sigma^3} = \frac{E[X - E(X)]^3}{(\text{var } X)^{\frac{3}{2}}}, \text{ kde} \quad (4)$$

μ je centrální moment, σ je směrodatná odchylka, $E(X)$ je střední hodnota a $\text{var } X$ je rozptyl

Koeficient špičatosti (Kurtosis)

Koeficient špičatosti je označován jako γ_2 a charakterizuje rozdělení náhodné veličiny vzhledem k normálnímu rozdělení. Určuje, jak se v rozložení četnosti vyskytují velmi vysoké nebo velmi nízké hodnoty. Rozdělení s koeficientem špičatosti 0 lze označit jako normální. Od tohoto se odvozuje další rozdělení. Pokud je špičatost kladná, jedná se o rozdělení špičatější oproti normálnímu rozdělení a v případě záporné špičatosti o rozdělení plošší oproti normálnímu rozdělení.

$$\gamma_2 = \frac{\mu_4}{\sigma^4} = \frac{E[X - E(X)]^4}{(\text{var } X)^2}, \text{ kde} \quad (5)$$

μ je centrální moment, σ je směrodatná odchylka, $E(X)$ je střední hodnota a $\text{var } X$ je rozptyl

KPI indikátor (Key performance indicator)

KPI, z anglického Key performance indicator je nástroj, který pomáhá k jednoduchému vyjádření výkonnosti nebo vzájemnému porovnání dat. Nejčastěji slouží pro vyjádření míry úspěšnosti definovaného procesu, například míra udržení obratu společnosti nebo výkonnosti. Definice KPI indikátoru vychází z nastavených cílů, vůči kterým je aktuální hodnota porovnávána. Výsledkem KPI indikátoru je číslo, nejčastěji uváděné v procentech (0 – 100 %), které vyjadřuje, jak moc se plán dokáže plnit.

Použití odchylky od „zdravého“ stavu

Principu KPI je využito při návrhu indikátoru, který reprezentuje stav robotu za daný časový interval. Hodnota KPI nabývá hodnot 0 – 100 % a reprezentuje stav robotu vzhledem k jeho „zdravému“ stavu, kdy 100 % je považováno za nejlepší stav a 0 % za nejhorší. Jakožto porovnávané hodnoty si lze představit průměrnou hodnotu proudu robotu za jeden programový cyklus. Při výpočtu je aktuální hodnota srovnána s hodnotou naměřenou za „zdravého“ stavu a výsledkem je KPI indikátor, který vyjádří podobnost zpracovávaného intervalu vzhledem ke „zdravé“ hodnotě. Hodnota ve „zdravém“ stavu je experimentálně ověřena na novém robotu a slouží jakožto etalon pro následné porovnávání aktuálních hodnot za provozu robotu.

$$KPI = \frac{H_0}{H_Z} \cdot 100, kde \quad (6)$$

H_0 je okamžitá hodnota,
 H_Z je hodnota ve zdravém stavu

Algoritmy pro predikci stavu

Algoritmy pro predikci stavu slouží pro předpovězení možného vývoje trendu. V této práci jsou algoritmy použity na aproximaci hodnoty KPI reprezentující stav robotu. Jejich úkolem je na základě tendence analyzovaných hodnot dokázat předpovědět budoucí stav a určit, kdy dojde k poklesu KPI pod kritickou úroveň. Úroveň může reprezentovat čas, kdy již bude nutné provést servisní zásah, nebo jiné události reagující na zhoršující se stav robotu.

Pro přehledné zobrazení trendu vývoje KPI v čase je použita aproximace za pomoci polynomu. Principem je výpočet koeficientů interpolačního polynomu x -tého stupně pro páry hodnot X, Y za pomoci metody nejmenších čtverců. Výsledkem je vektor obsahující koeficienty. Výpočet je realizován v nástroji ibaAnalyzer za pomoci funkce LSQPolyCoef a vykreslení aproximační přímky za pomoci funkce Polynomial. Příklad rovnice polynomu 2. řádu:

$$y = ax^2 + bx + c \quad (7)$$

Úroveň řádu polynomu lze v ibaAnalyzer funkcích velmi jednoduše upravit a docílit proložení křivkou vyššího řádu. Při volbě řádu je nutné myslet na fakt, jak velké proložení je žádané. Zda se má jednat o přímku reprezentující vývoj hodnot nebo o více řádový polynom, který více kopíruje sledovaný signál.

4.3 Sběr dat

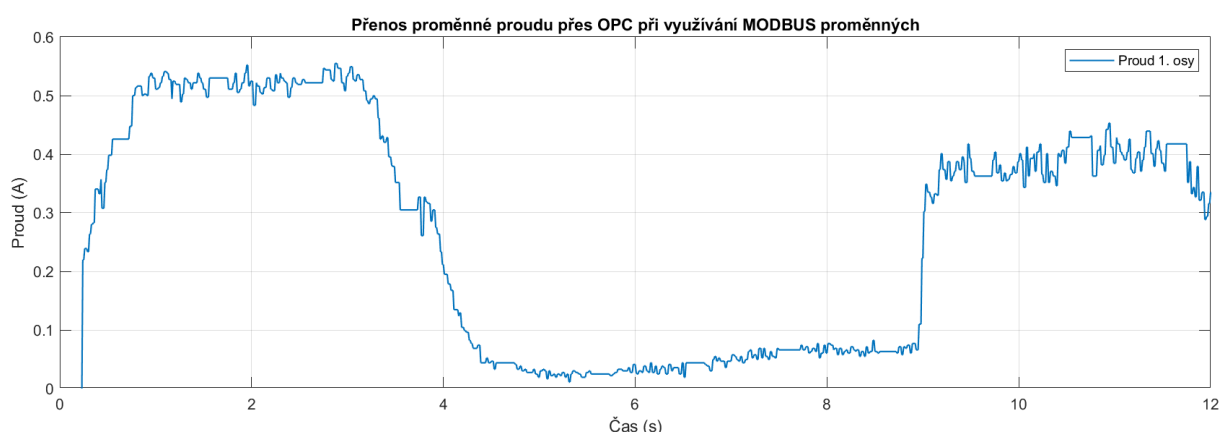
Sběr dat z robotického pracoviště je základním pilířem pro analýzu průběhů provozu robotu a budování systému prediktivní údržby. Robot nabízí několik komunikačních rozhraní, pomocí kterých lze data přenášet a několik řešení odkud budou samotná data zaznamenávána a ukládána. Potřebná data jsou uložena v proměnných robotu a je potřeba provést extrakci dat do místa sběru.

Za pomoci testování a vzhledem k hardwarové výbavě robotu byla pro sběr dat vybrána komunikace OPC-UA. Jedná se o relativně rychlou a v robotu dobře přístupnou komunikaci umožňující se dostat k potřebným proměnným robotu. Tato komunikace bude realizována mezi robotem a řídicím systémem na PLC, aby byly aktuální data o provozních hodnotách dostupná také pro systém řízení. Pro řízení robotu je již realizovaná komunikace za pomoci Modbus protokolu, která slouží pro spouštění sekvencí z PLC a čtení informací o stavu robotu. Další data, která je nutno zaznamenat, jsou data z externích senzorů, které jsou připojeny a napojeny do řídicího PLC. Jedná se o měření proudů jednotlivých os robotu, které je realizováno za pomoci měřicích cívek a připojení na analogové vstupy řídicího PLC. Sjednocením všech dat v PLC je zajištěn přístup k datům z jednoho místa a možnost použití dat také interně pro řízení robotu. Z PLC budou data dále zaznamenávána za pomoci softwaru ibaPDA, který nabízí přímou komunikaci se Siemens PLC. Pomocí zvolení konkrétních zaznamenávaných signálů dojde k definici signal listu a zahájení záznamu s ukládáním dat do souboru. Z pohledu PLC není pro tento záznam nutno žádného programování, pouze zajištění adresního (neoptimalizovaného) přístupu k datovým strukturám a povolení čtení dat.

4.3.1 Robotická data

Robot nabízí zásadní data potřebná pro realizaci systému prediktivní údržby. Jelikož jsou veškerá data shromažďována v PLC, z kterého až následně probíhá záznam a zpracování, je nutno zvolit vhodné komunikační rozhraní mezi robotem a PLC. Z robotu je nutné pro systém prediktivní údržby zaznamenávat data především o aktuálních proudech jednotlivých os, ale také další stavové proměnné robotu. Pro řízení robotu z PLC je použita komunikace Modbus, která přenáší data přibližně každou jednu sekundu (co 500 ms je střídán zápis a čtení dat). Z provedeného testování dochází k častému zpoždění komunikace a čas pro výslednou aktualizaci proměnných se ještě prodlužuje. To je pro použití z hlediska záznamu aktuálního proudu na osách či dalších provozních hodnot robotu naprosto nedostatečné. Při sledování stavu robotu je nutné reagovat na proudové špičky a další jevy, které rychlou komunikaci vyžadují. Jelikož robot, na kterém je práce prováděna, obsahuje pouze omezená komunikační rozhraní, došlo k realizaci vyčítání dat za pomoci protokolu OPC UA. Při porovnání s dnešními moderními komunikačními standardy se nejedná o nejrychlejší řešení, ale použití jiného rozhraní nebylo možné z důvodu SW a HW výbavy pracoviště. Z dostupných komunikačních rozhraní bylo OPC UA nejrychlejší.

Roboty Fanuc obsahují vestavěný OPC UA server, který je přístupný ze sítě robotu a dostupný na adrese `opc.tcp://ROBOT_IP:4880/FANUC/NanoUaServer`. Jak již bylo uvedeno v předešlých kapitolách, server nabízí nodes (uzly), které poskytují data o robotu. Hlavním zdrojem dat je node „RobotInformation“, z kterého jsou čteny aktuální hodnoty proudů jednotlivých os. Dokumentace Fanucu definuje minimální dobu pro čtení hodnot ze serveru na 100 ms. Při realizaci řady pokusů a využívání pouze OPC UA komunikace došlo k realizaci čtení dat každých 20 ms, což je výrazně rychleji než uvádí výrobce. Interval pro čtení dat za pomoci OPC UA protokolu se výrazně zpomalil ve chvíli, kdy byly čteny data z Modbus struktury, která je rovněž přístupná přes OPC UA komunikaci. Všechny nodes z Modbus struktury robotu jsou velikostně 10 000 až 16 000 bytů velké a bez možnosti indexování jejich konkrétních částí dochází k odesílání celé velikosti pole dat. Odeslání celého node má zásadní vliv na přenosovou rychlost a komunikace ostatních proměnných zcela ustane, jelikož je komunikace zaneprázdněna přenosem velkého množství dat.



Obrázek 13 – Přenos proměnné proudu přes OPC při využívání Modbus proměnných

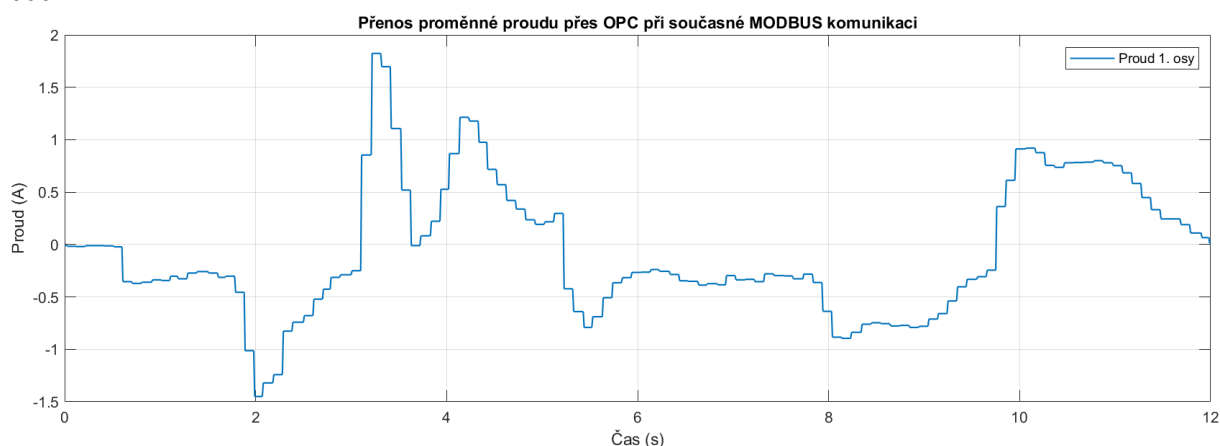
Z pohledu realizace OPC UA klienta na straně PLC nelze definovat omezený datový rozsah čtení node struktury. Data, která jsou v daných Modbus strukturách uložena, jsou tedy velmi obtížně přenositelná na PLC. Při přenosu proudů za pomoci OPC komunikace každých 20 ms došlo při odesílání Modbus struktury k okamžitému pozastavení komunikace až po dobu jedné sekundy. Řešením bylo realizovat

vyčítání Modbus dat pouze v delších intervalech a nezatěžovat tak komunikaci důležitých okamžitých hodnot proudů. Pro vylepšení rychlosti komunikace byly odděleny části přenosu „rychlých“ dat proudů a „pomalých“ provozních dat z Modbus struktury. Obě části stále běžely na OPC UA komunikaci, ale „rychlá“ data byla vyžadována každých 20 ms a „pomalá“ data pouze jednou za vteřinu. Tím došlo k minimalizaci zatěžování komunikace, ale stále při přenosu velkého množství dat přes Modbus strukturu docházelo k prodlevě 200 ms. Na průběhu (Obrázek 13) lze pozorovat rychlý přenos proudu osy při periodě obnovy 20 ms, ale co jednu sekundu při přenosu velké Modbus struktury dojde k pozastavení komunikace.

Komunikace tedy přenášela více hodnot o proudu, ale zřetelně lze pozorovat místa, kde docházelo ke komunikaci Modbus struktury. Pomalejší komunikací zároveň výrazně vzrostla časová nepřesnost Modbus dat, která mohla dosahovat zpoždění až 1,2 s. Z pohledu následného zpracování proudů v závislosti na čísle programu z Modbus struktury docházelo k výrazným nepřesnostem. Na základě provedeného testování, byla Modbus struktura kompletně vyjmuta z OPC komunikace a proměnné, které měly být přes tuto komunikaci posílány budou posílány přímo přes Modbus komunikaci.

4.3.2 Řídící proměnné PLC

Jak je uvedeno v předchozí kapitole, čtení Modbus dat za pomoci OPC UA protokolu způsobuje velmi výrazné zpomalení komunikace mezi robotem a PLC. Data, která měla být touto cestou přenášena, jsou provozní veličiny robotu jako operační stav robotu, run, fault nebo číslo běžícího programu. Tato data však lze získat také za pomoci přímé Modbus komunikace, která je využívána při řízení robotu. Některé proměnné jsou již v rozhraní mezi robotem a PLC obsaženy a další je možné do rozhraní přidat. Výhodou Modbus komunikace oproti integraci jeho data modelu do OPC UA, je možnost čtení konkrétního rozsahu dat, takže nebude docházet k přenosu velkého množství dat a výraznějšímu zpomalování komunikace. K jistému zpomalení komunikace docházet bude, ale není tak zásadní jako při minulém testu. Při paralelním provozu OPC komunikace (20 ms) a Modbus komunikace (500 ms) došlo k získávání aktuálních hodnot proudu na osách v intervalech 80 – 100 ms. Rychlost OPC komunikace se lehce zpomalila, ale došlo k eliminaci výpadků vzniklým přenosem velkého množství dat.



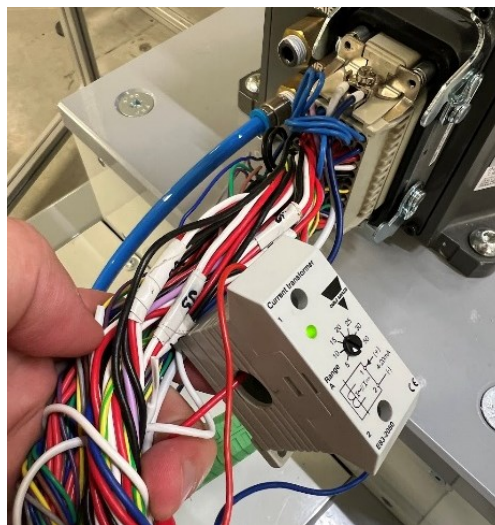
Obrázek 14 – Přenos proměnné proudu přes OPC při současné Modbus komunikaci

4.3.3 Měření proudu robotu, volba vhodného hardwaru

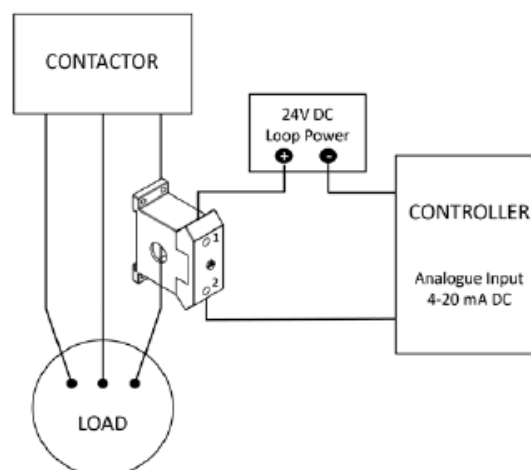
Jak již bylo zmíněno v předchozích kapitolách práce, robotické manipulátory jsou vybaveny omezenými možnostmi, co se týče komunikačních rozhraní a protokolů. Žádný z nich nenabízí realtime, nebo rychlé

komunikační rozhraní, které by nabízelo čtení aktuálních hodnot v rámci jednotek ms. Snahou práce je návrh systému, který by byl univerzální a jednoduchý pro nasazení na roboty různých výrobců. Z těchto důvodů došlo k návrhu a realizaci vlastního měření proudů os na přívodním kabelu robotu. Při návrhu bylo uvažováno s dvěma možnostmi použití měřicích AI karet a dvojicí měřicích transformátorů proudu. Součástí práce bylo testování možných komponentů a určení, která kombinace měřicích karet je pro požadavky prediktivní údržby vhodnější.

Měření probíhalo experimentálně na motoru první osy robotu, který je napájen za pomoci tří fázových vodičů. Za předpokladu, že motor je fázově vyvážený, došlo k umístění měřicích karet na libovolný fázový vodič motoru. Měřicí transformátor od výrobce Carlo je průvlakový a bylo ho nutné navléknout na přívodní vodič motoru. Trafo je napájeno přímo z napěťové smyčky měřicí karty, stačilo tedy pouze napojení na měřicí kartu. Trafo nabízí pevný proudový analogový výstup a to 4-20 mA, je tedy nutné připojení na vhodnou analogovou kartu a správné nastavení kanálu.



Obrázek 15 – Zapojení trafo Carlo



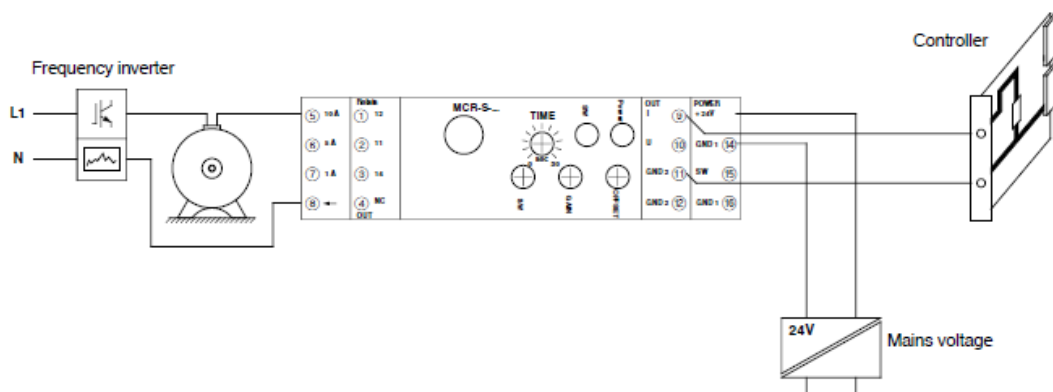
Obrázek 16 – Schéma zapojení serva Carlo [25]

Druhý měřicí transformátor PXC byl umístěn na druhou fázi motoru první osy robotu. V tomto případě bylo nutné vodič napojit na příslušné svorky měřicí karty. Odhadovaný rozsah, který by mohl být na motoru robotu změřen byl do 5 A, kabel byl tedy připojen na vstupní svorky odpovídající 5 A.

Pro napájení byla karta připojena na zdroj 24 V, jelikož oproti druhé měřicí kartě pro svůj provoz vyžaduje externí napájení. Výstupem karty byl zvolen proudový rozsah 4-20 mA.



Obrázek 17 – Zapojení trafo PXC



Obrázek 18 – Schéma zapojení trafo PXC [24]

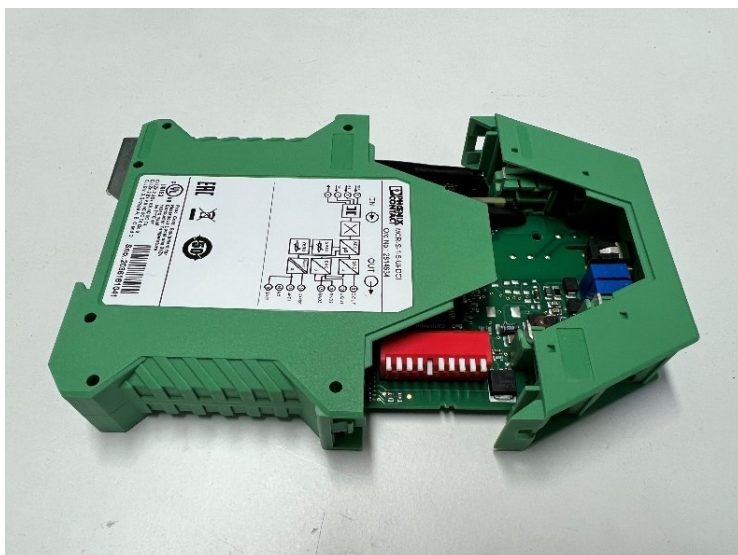
Jelikož karta nabízí širokou variaci nastavení, po připojení bylo nutné definovat jaké rozsahy bude karta měřit, jakou veličinu bude měřit, jaká je výstupní veličina, jaký je výstupní rozsah a další parametry měření včetně doladění rozsahů na potenciometrech. Nastavení je prováděno za pomoci přepínačů, umístěných pod ochranným krytem karty. Jedná se o deset přepínačů definujících výše zmíněné parametry. Nastavení pro měření bylo následovné:

Tabulka 12 – Nastavení přepínačů PXC karty

Přepínač	1	2	3	4	5	6	7	8	9	10
Stav	OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	OFF

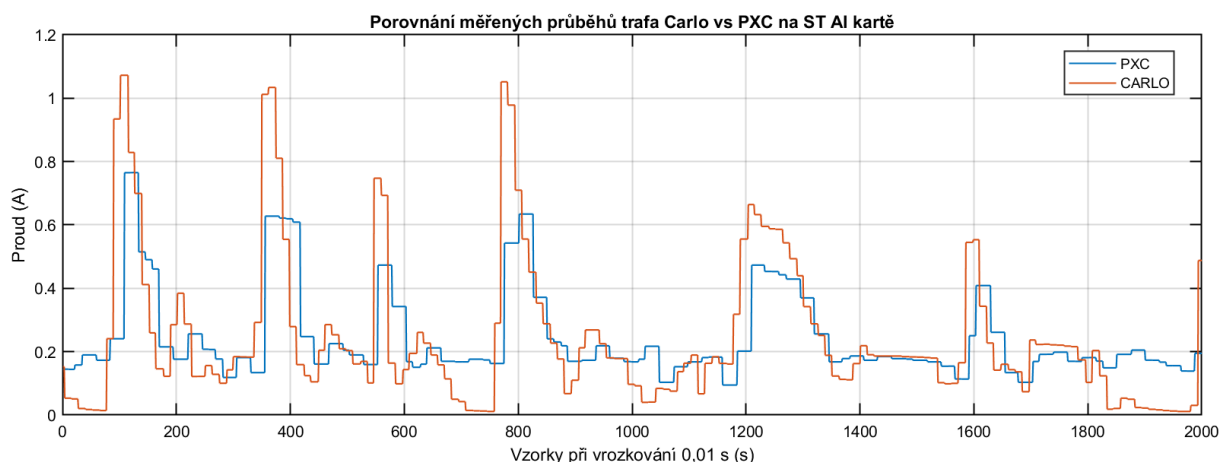
Shrnutí provedeného nastavení:

- Měřený rozsah 0 A ... 5.00 A na 5 A svorkách
- Proudový výstup karty 4 mA ... 20 mA
- Provoz nad threshold hodnotou
- Měření True r.m.s. hodnoty (AC and DC without without sign bitdetection)
- Konfigurace karty za pomoci switchů a nikoliv programově



Obrázek 19 – Nastavení přepínačů PXC karty

Při prvním testovacím měření byly karty připojeny dvou vodičově na analogovou kartu AI 4x U/I 2-wire ST. Čas cyklu analogové karty závisí na počtu aktivních kanálů a proto byly aktivovány pouze 2 potřebné kanály a zbylé byly deaktivovány. Integrace byla nastavena na 50 Hz, tedy vzorkovací frekvence jednoho kanálu byla 60 ms a celkový cycle time karty byl 120 ms. Jelikož se jednalo o standard „ST“ kartu, nebylo možné dosáhnout rychlejších průběhů.

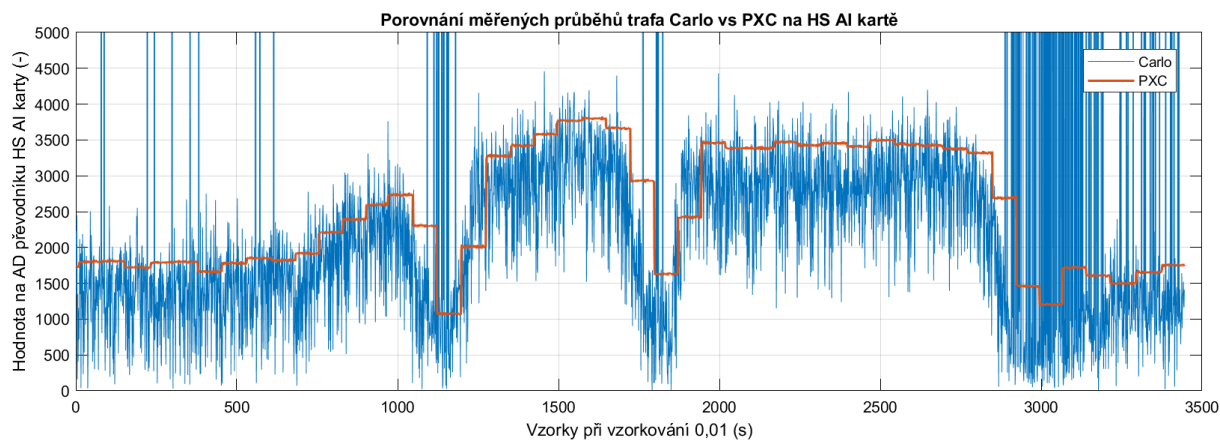


Obrázek 20 – Porovnání měřených průběhů trafo Carlo vs PXC na ST AI kartě

Z grafu lze pozorovat, že měřicí trafo Carlo (červený průběh) dokáže zaznamenat větší špičky, než měřicí trafo od Phoenix contactu (modrý průběh). Za pomoci markerů byla změřena nejmenší obnovovací frekvence průběhu, která dle předpokladu a parametrů analogové karty PLC byla 120 ms. V porovnání se získáváním dat za pomoci OPC UA protokolu, který je schopen v ideálním případě komunikovat každých 20 ms, došlo k výraznému zmenšení vzorkovací frekvence. Měřením došlo k ověření, že rychlosti standardní AI karty pro měření aktuálních průběhů proudů nedostačují a je třeba použití rychlejší měřicí PLC AI karty.

Pro další testy byla k PLC přidána rychlejší analogová karta AI 2x U/I 2-/4-wire HS a proudová trafo přepojena do této nové karty. Trafo byla zapojena shodně jako na předchozí kartě a to dvojvodičově. Z parametrů karty je patrný velmi výrazný nárůst obnovovací frekvence karty. Obnovovací čas jednoho kanálu karty je 10 μ s a celkový obnovovací čas celé karty se všemi

aktivovanými kanály je 125 μ s. To je obrovský nárůst oproti „ST“ kartě, která se obnovovací frekvencí dostala maximálně na 120 ms. Po připojení byla provedena obdobná sekvence na robotu, aby byla ověřena funkčnost nového zapojení.

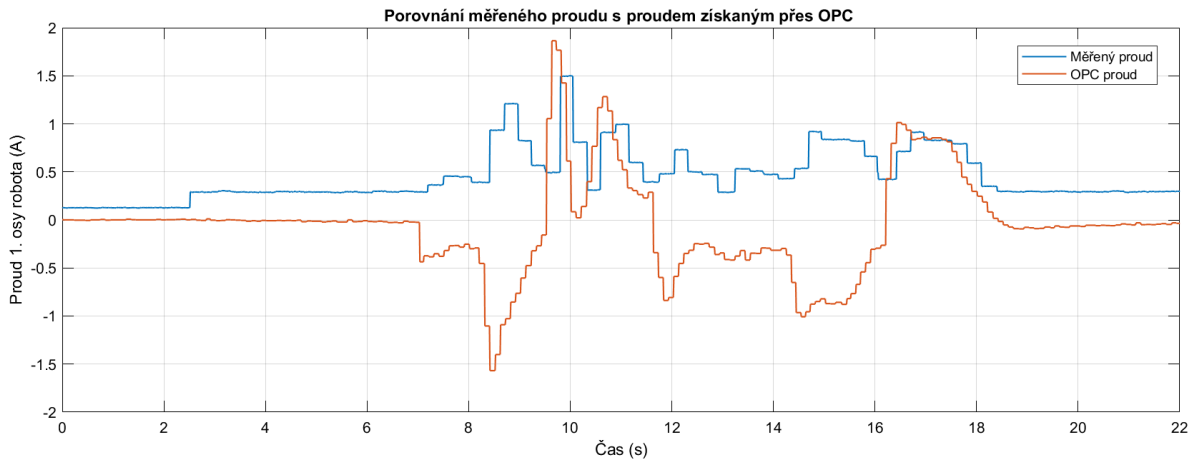


Obrázek 21 – Porovnání měřených průběhů trafo Carlo vs PXC na HS AI kartě

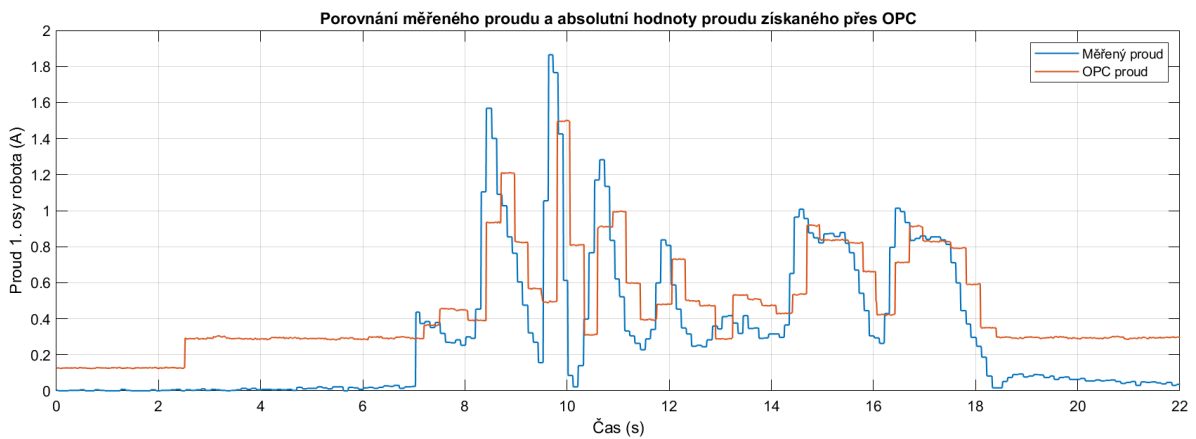
Na grafu lze pozorovat porovnání měřicích transformátorů Karlo a PXC při připojení na vysokorychlostní analogovou kartu. Frekvence záznamu dat se dostala na 3,8 ms. Karta PXC zaznamenává průběh věrohodně, ale průběh z karty Karlo obsahuje velké množství šumu. Z průběhu lze vidět, že trafo Carlo reaguje na průběh proudu, jenže jeho výstup je zašuměn do takové míry, že nelze určit přesnou hodnotu měření. Při měření došlo ke snaze zamezení vzniku rušení na kabelu, přepojení na jiný kanál, aplikace smoothingu, ale žádné ze zmíněných opatření nemělo na míru šumu vliv. Výstup na AI PLC kartu je realizován za pomoci proudové smyčky, a měl by tedy být odolný vůči rušení. Signál by bylo možné aproximovat do přesnější křivky, došlo by ale ke ztrátě informací o špičkových proudech, které jsou při měření velmi důležité. Při pohledu na průběh karty PXC lze pozorovat pomalou odezvu na měřený signál proudu. I přes dlouhodobé a podrobné testování vzniku tohoto „obdélníkově“ tvarovaného signálu, došlo k zjištění, že se jedná o reálný průběh vracený samotnou kartou. Rychlost obnovy hodnoty na výstupu je přibližně 280 ms, což odpovídá údajům o kartě, definující dobu odezvy karty na skokovou změnu měřené veličiny na 330 ms. Bohužel vysokým časem odezvy na měřící veličinu dochází ke ztrátě informací o špičkách. Stále se ale jedná o lepší průběh, než bylo schopné poskytnout měřící trafo Carlo. Pro další měření proudu na přívodních kabelech robotu byla použita kombinace high speed AI karty v PLC a měřícího proudového trafo PXC.

Pokud dojde k porovnání hodnot proudů osy měřených za pomoci proudového trafo PXC a hodnot proudů os získaných za pomoci OPC komunikace, lze pozorovat několik rozdílů. Hlavním rozdílem je, že měřený proud na kabelu robotu je měřen absolutně a bez znaménka. Oproti tomu proudy z OPC komunikace jsou se znaménkem a lze z nich určit směr pohybu robotu. Proud měřený na přívodním kabelu je okamžitý RMS proud odebíraný osou. Proud získávaný z OPC komunikace je v robotu označován jako „Torque“ a v dokumentaci popisován jako Q fázový proud v ampérech. Jedná se o proud, který je potřebný k vyvinutí točivého momentu osy. Z grafu lze také pozorovat neshodu mezi měřeným a Q proudem hned při začátku pohybu. U Měřeného proudu lze pozorovat skok na 0,3 A, který nastal při spuštění programu robotu, tedy došlo k jisté přípravě motorů na pohyb. Q proud v robotu to ovšem neovlivnilo a zůstává stále na nule. Již z tohoto důvodu nelze oba proudy vzájemně porovnávat a nelze je považovat za shodné. Při měření obou proudů zároveň bude tvar průběhu

shodný, ale hodnotově se budou lišit. To lze pozorovat při provedení překlopení záporné části OPC proudu (Q proudu robotu) a porovnáním s měřeným proudem z PXC měřicího transformátoru.

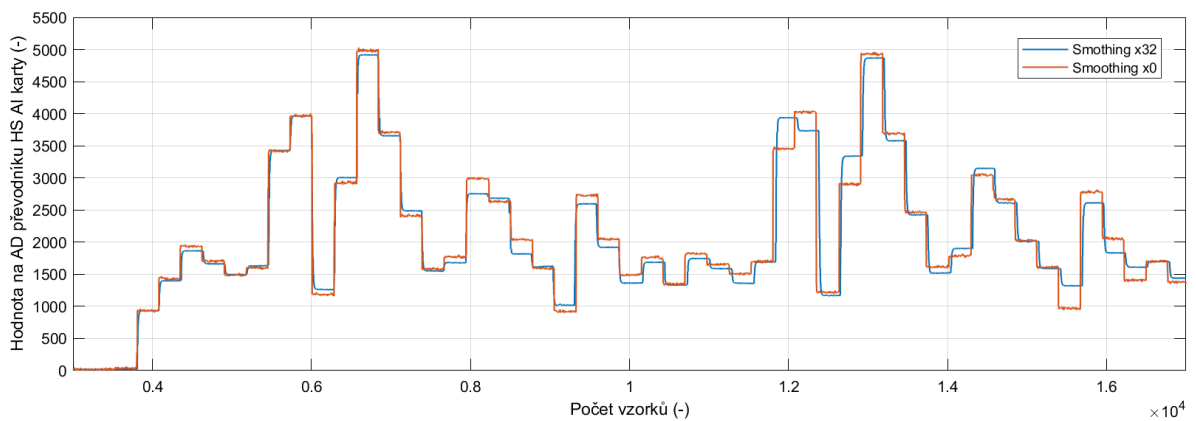


Obrázek 22 – Porovnání měřeného proudu s proudem získaným přes OPC



Obrázek 23 – Porovnání měřeného proudu a absolutní hodnoty proudu získaného přes OPC

Během měření došlo k testování aplikace smoothingu na analogové kartě PLC, výsledkem kterého je proložení křivky zaznamenaného signálu a vyhlazení výsledné křivky. Pro otestování byl zaznamenan stejný průběh robotu při aplikaci smoothingu 0x a 32x. Z grafu lze pozorovat vyhlazení hran skokových změn. Tento jev není pro výsledky měření jinak prospěšný, a proto byla veškerá další data měřena při smoothingu 0x.



Obrázek 24 – Porovnání aplikace smoothingu na analogové kartě PLC

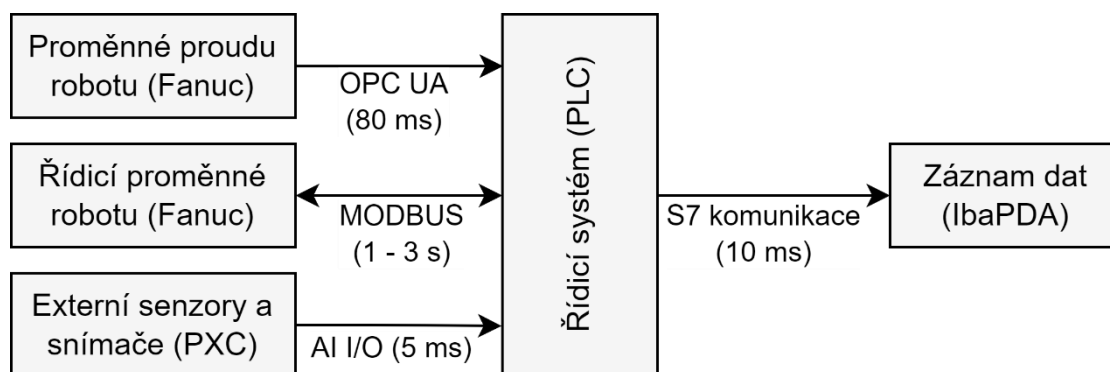
5 Realizace navrženého systému monitorování robotu

Praktická realizace implementace prediktivní údržby stavu pracoviště byla prováděna na výše popisovaném robotickém pracovišti. Pracoviště obsahuje dvojici robotických manipulátorů, které jsou hlavním sledovaným objektem. Konkrétně se jedná o jejich servomotory, které jsou dle následujícího postupu monitorovány a detekovány změny v jejich chování. Součástí monitorování je archivace dat, implementace analýz stavu a použití algoritmů pro predikci budoucího stavu pracoviště. Implementaci systému sledování stavu robotu lze rozdělit do několika bodů, reprezentující jednotlivé kroky, které je nutno pro dosažení žádaného výsledku provést. Každá kapitola reprezentuje určitou operaci s daty a navazuje na operaci předchozí.

Praktická část práce je realizována na robotu Fanuc LR Mate 200iD. Oba roboty jsou však totožné a lze použít stejných postupů a principů pro rozšíření systému monitorování také na druhého robota. Z důvodu testování možností měření a omezené dostupnosti hardwarových komponentů je realizováno měření proudů pouze pro první osu robotu. Pro záznam a zpracování dat je využíváno softwarů od firmy iba, konkrétně ibaPDA pro sběr dat a ibaAnalyzer pro zpracování dat. Nástroje umožňují také nahrání dat do databáze, která slouží jako místo pro ukládání dat a jejich poskytnutí pro vizualizační systém. Pro vizualizaci dat byl zvolen open source nástroj Grafana. Pro účely testování a prezentace funkčnosti navrženého systému byla definována pohybová sekvence robotu, která reprezentovala obdobné pohyby, jako jsou prováděny při skutečném provozu robotu. Veškeré grafy a výsledky jsou směřovány k této pohybové sekvenci.

5.1 Sběr dat

Cílem sběru dat je shromáždění všech dostupných a užitečných dat, která lze pro prediktivní údržbu využít. Cílovou lokací sběru dat jsou datové bloky v PLC programu, odkud budou následně data dále v řetězci čteny. Sběr dat je realizován za pomoci tří postupů. Komunikace s OPC serverem robotu, měření za pomoci analogových karet a vyčítání dat z již existující řídicí Modbus komunikace mezi PLC a robotem. Data jsou shromažďována v PLC především z důvodu jednotného záznamu za pomoci jednoho zařízení a také z důvodu možnosti realizace zpracování dat přímo v PLC. Zpracování v PLC nebude při této práci realizováno, ale jedná se o možnost dalšího rozšíření této práce. Proces sběru a záznamu dat je popsán blokovým schématem na následujícím obrázku (Obrázek 25). Lze pozorovat poměrně velké rozdíly mezi rychlostmi komunikace, která je klíčovou složkou pro realizaci algoritmů sledování stavu robotu. Rychlosti komunikačních rozhraní byly podrobně analyzovány v předchozí kapitole. V této kapitole jsou již použita nejlepší (nejrychlejší) řešení pro komunikaci jednotlivých proměnných, která vychází z variant testování.



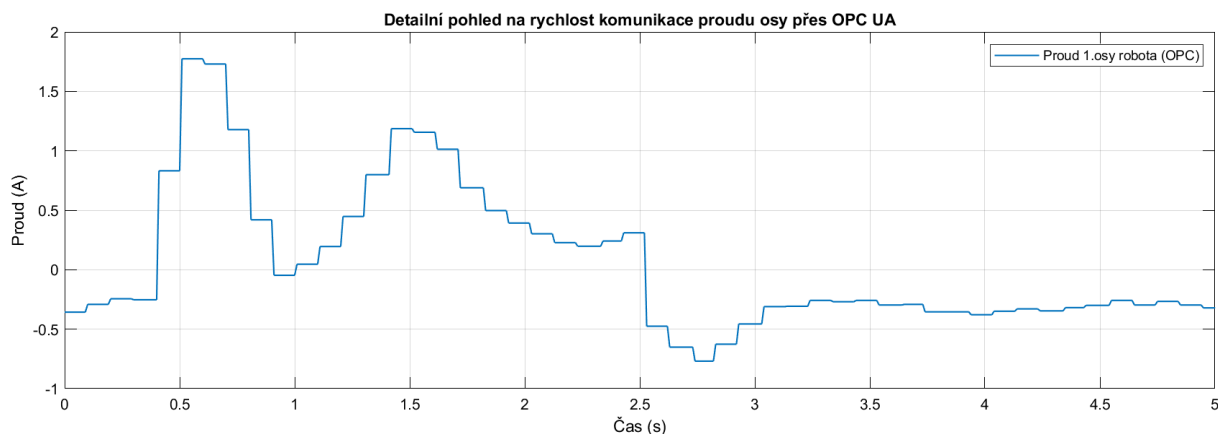
5.1.1 Robotická data

První skupinou zpracovávaných dat jsou data dostupná pro čtení z řídicí jednotky robotu. Jejich přenos do PLC s dodržení nejvyšší možné frekvence vzorkování není tak jednoduchý, jak se na první pohled může zdát. Zároveň jsou data v robotu dělena do mnoha vnitřních proměnných a samotná příprava v robotu je náročná. Obecně lze data z robotu přenášet za pomoci dvojice komunikačních protokolů, kterými je Modbus a OPC-UA. Ideální cestou by bylo posílání všech dat pomocí jedné komunikace, která by byla dostatečně rychlá a výkonná pro odeslání všech potřebných proměnných. To ale na dostupném pracovišti není možné a je nutné data rozdělit podle možností robotu. Modbus nabízí možnost komunikace všech dostupných dat v robotu, za pomoci vnitřního programu se daná proměnná zapisuje na výstup robotu, odkud je čtena v PLC. Nevýhodou je velmi pomalá rychlost komunikace v rámci 500 ms až 1 s. Přenos dat za pomoci OPC-UA komunikace je přesný opak. Nabízí rychlé odesílání dat v rámci 20 – 100 ms, ale touto rychlostí je možné číst pouze přednastavené proměnné. Definice vlastních odesílaných proměnných je možná, ale realizována za pomoci Modbus data modelu. Problémem Modbus data modelu je nutnost do PLC odeslat celý datový blok, který je natolik velký, že dojde k zastavení komunikace po dobu odesílání, a tedy k výraznému zpomalení celkové komunikace. Z tohoto důvodu jsou přes OPC komunikaci čteny pouze proměnné dostupné ve výchozím nastavení, jako jsou proudy os robotu, které jsou primární sledovanou veličinou a je nutný jejich záznam za pomoci nejvyšší možné vzorkovací frekvence. Další uživatelsky definované proměnné jsou odesílány pomocí pomalejší Modbus komunikace.

Tabulka 13 – Odesílané proměnné robotu přes OPC-UA

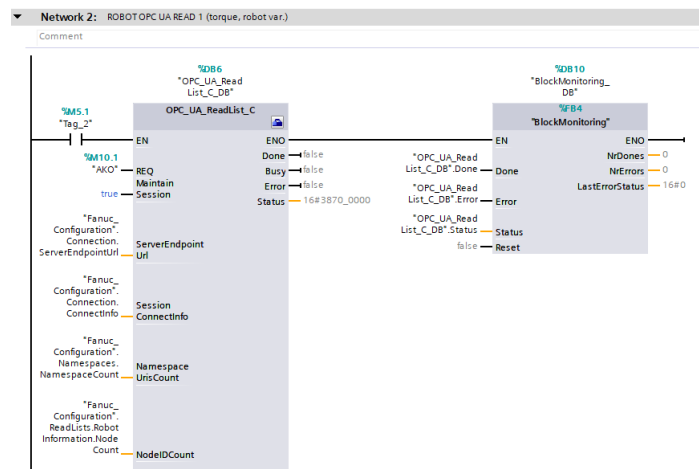
Proměnná	Datový typ	Cesta na OPC-UA serveru	Popis
Torque	Float[6]	Objects/Server/Torque	Pole okamžitých Toque (Q proud) hodnot jednotlivých os robotu
ProgramSpeed	Byte	Objects/Server/ ProgramSpeed	Aktuální rychlost robotu

Při měření rychlosti komunikace je nutné dbát také na souběh obou komunikací, který může ovlivňovat jejich rychlost. Samotná OPC komunikace byla schopna odesílat data o aktuálních proudech na motorech každých 20 ms. Při souběžném běhu s Modbus komunikací tato rychlost klesla na 80 – 100 ms. Stále se však jedná o přijatelnou rychlost, oproti odesílání dat za pomoci přímé Modbus komunikace nebo za pomoci Modbus data modelu přes OPC.



Obrázek 26 – Detailní pohled na rychlost komunikace proudu osy přes OPC UA

OPC UA server běží na robotu a PLC je v komunikaci jeho klientem. Pro připojení a čtení dat je využito funkce OPC_UA_Read_List v TIA portalu. Data jsou po serveru žádána každých 20 ms a je sledováno, zda dojde k jejich přijetí. V případě zatížení komunikace jsou obdrženy data v nejkratším možném intervalu.



Obrázek 27 – Program pro OPC UA komunikaci

Další data z robotu jsou do PLC odesílána pomalejší Modbus komunikací. U těchto dat už tolik nezáleží na rychlosti jejich čtení a lze Modbus komunikaci využít. Jedná se o provozní data robotu, která slouží pro identifikaci aktuálního stavu robotu. Příkladem jsou proměnné Busy, Exe, ProgRung, Rung a další. Tyto proměnné jsou nutné pro rozpoznání intervalu běhu analyzované sekvence, tedy aby byl vyhodnocován pouze proud motorů při sledovaném pohybu. Podrobnosti o prováděné analýze jsou specifikovány v dalších kapitolách.

Tabulka 14 – Proměnné čtené z robotu za pomoci Modbus komunikace

Proměnná	Datový typ	Cesta v robotu	Popis
Program	Int	GO[1], DO[33...48]	Číslo vykonávaného programu
Temp_J1	Int	GO[2], DO[97...124]	Teplota motoru 1. osy robotu
ProdRung	Bool	DO[11]	Robot má spuštěnou produkci
Busy	Bool	DO[15]	Robot vykonává program
Rung	Bool	DO[1]	Robot vykonává konkrétní program
Done	Bool	DO[0]	Robot dokončil konkrétní program
RobReady	Bool	DO[7]	Robot připraven ke spuštění programu

Kromě proměnných čtených z robotu, došlo také ke čtení proměnných přímo z PLC. Tyto proměnné jsou do robotu odesílány a není nutná jejich zpětná komunikace, ale pouze vyčítání z komunikační struktury.

Tabulka 15 – Proměnné odesílané do robotu za pomoci Modbus komunikace

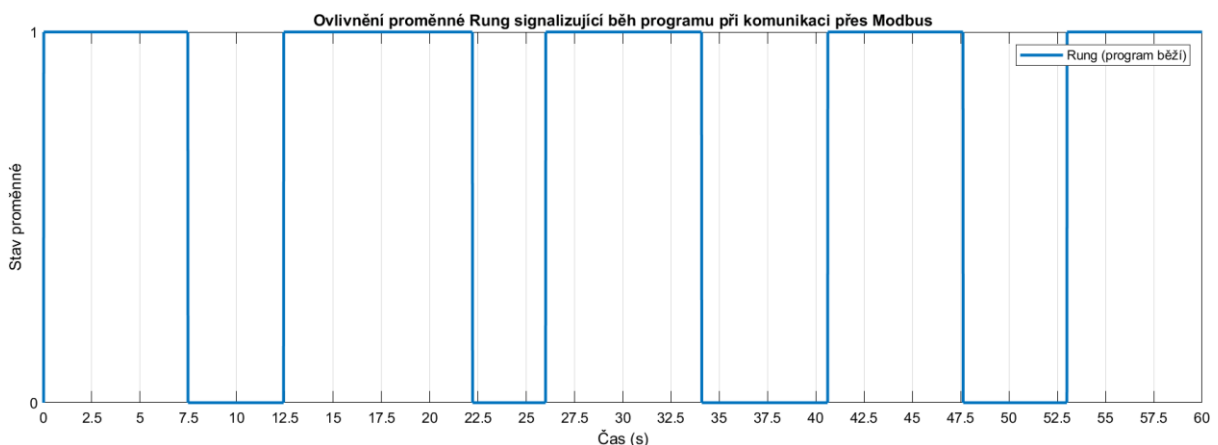
Proměnná	Datový typ	Cesta v robotu	Popis
Program	Int	GI[1], DI[33...48]	Číslo programu robotu
Exe	Bool	DI[0]	Spuštění programu robotu
Enb	Bool	DI[4]	Aktivace motorů robotu

Modbus komunikace je řízena z PLC za pomoci komunikačního bloku a logiky programu. Komunikace probíhá v cyklech 500 ms a dochází ke střídání čtení a zápisu dat do robotu. Pro sběr dat je využito pouze čtení, které probíhalo jednou za 1 sekundu. V reálném použití ale docházelo k většímu prodloužení času komunikace až k 2 – 3 sekundám. Nestálost komunikace lze pozorovat na následujícím příkladu (Obrázek 28), kde při stejné se opakujícím průběhu dojde k umělému prodloužení času běhu programu.

```
//Program na čtení a zápis dat do robotu za pomoci Modbus komunikace v TIA Portalu
#RqGetDataPE(CLK := "Clock_10Hz");

IF #RqGetDataPE.Q THEN
  #Comn.i := #Comn.i + 1;
  IF #Comn.i > 1 THEN
    #Comn.i := 0;
  END_IF;
END_IF;

// Střídání vyčítání a zápisu Modbus na náběžnou hranu co 0.5s
IF #Comn.i = 0 AND #RqGetDataPE.Q THEN
  #Comn.mode := 104;
  #Comn.addr := 0;
  #Comn.len := 16;
  #Comn.REQ := TRUE;
ELSIF #Comn.i = 1 AND #RqGetDataPE.Q THEN
  #Comn.mode := 116;
  #Comn.addr := 0;
  #Comn.len := 16;
  #RetVal := BLKMOV(SRCBLK := "Comn_DB".Snd, DSTBLK => "Comn_DB".AuxDATA);
  #Comn.REQ := TRUE;
ELSE
  #Comn.REQ := FALSE;
END_IF;
```



Obrázek 28 – Ovlivnění proměnné Ring signalizující běh programu při komunikaci přes Modbus

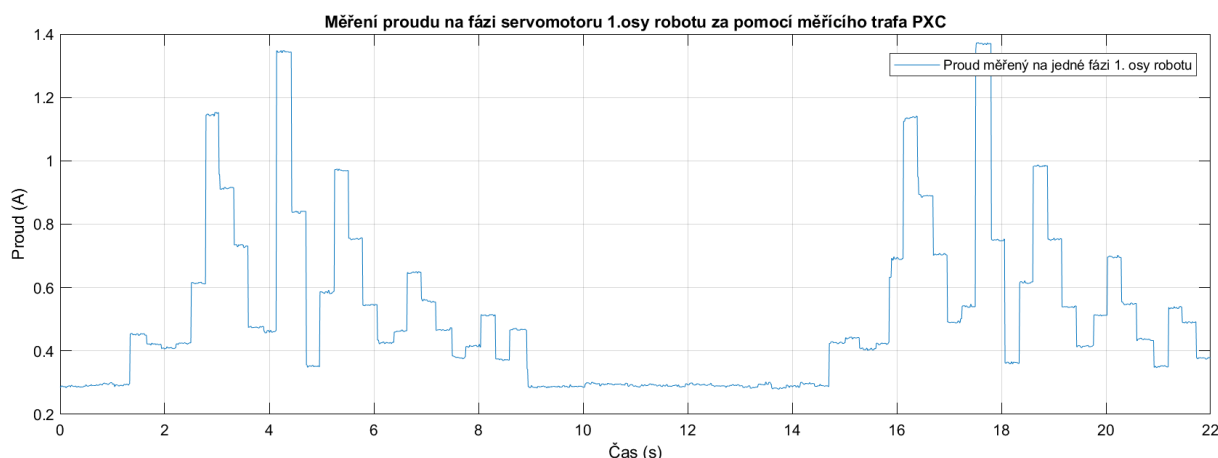
5.1.2 Data z měřicích karet

Z OPC UA komunikace jsou sice získávány údaje o prouděch jednotlivých os robotu, ale s obnovovací frekvencí přibližně 80 – 100 ms. Pro pomalé pohyby robotu se může zdát vzorkovací frekvence dostatečná, ale při použití vysokých rychlostí robotu může dojít ke ztrátě informací o proudových špičkách. Z toho důvodu bylo realizováno externí měření proudů na první ose robotu. Cílem je nabídnout variantu, která není závislá na proměnných robotu a možnost realizace externího systému prediktivní údržby. Součástí práce je také porovnání naměřených hodnot s hodnotami získanými za pomoci komunikace s robotem.

Pro měření proudů na přívodním kabelu robotu byla zvolena vysokorychlostní (HS) analogová karta, která disponuje časem cyklu pod 1 ms. Jakožto měřicí prvek byl vybrán měřicí proudový transformátor od výrobce Phoenix Contact. Podobný popis a testování variant měření je uveden v předchozí kapitole 4.

Tabulka 16 – Nastavení analogové karty pro měření proudu robotu

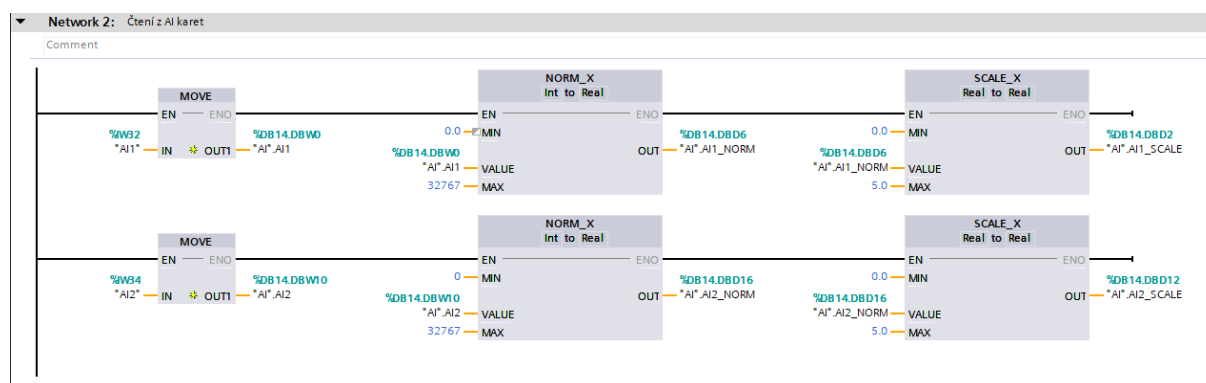
Parametr	Kanál	Rozsah	Smoothing
Hodnota	Current (2 wire)	4-20 mA	0



Obrázek 29 – Naměřený průběh na jedné fázi 1. osy robotu za pomoci měřícího trafo PXC

Z grafu lze pozorovat, že změna hodnot probíhá velmi pomalu (každých 200 ms) a jedná se o ještě pomalejší sběr dat, než při hodnotách z OPC UA komunikace. Rychlost je daná vlastnostmi měřicí karty, která vlivem vysokého času odezvy poskytuje sledovaný průběh. Měřicí karta je nastavena shodně jako při testech v předešlé kapitole, tedy měřený rozsah 0 – 5,00 A na 5 A svorkách a nastavený proudový výstup karty 4 – 20 mA. I přes ne příliš rychlý záznam aktuálních hodnot proudů na měřených vodičích se jedná o nejlepší výsledek, kterého jsem v práci byl schopen s příslušným hardwarem dosáhnout. Pro další zpracování bude pracováno s tímto průběhem a bude porovnáván s průběhem získaným z OPC.

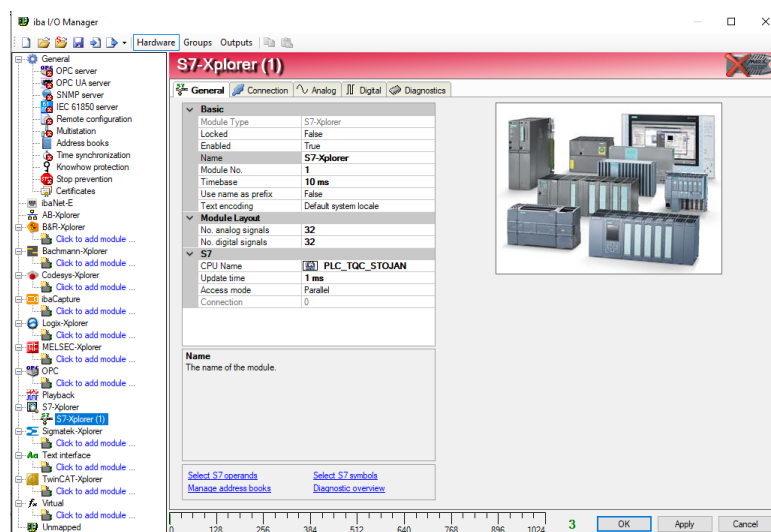
Převod dat dle rozsahu je v PLC programu realizován za pomoci funkcí NORM_X a SCALE_X. Po přečtení hodnoty z analogového vstupu dojde k normování na hodnotu 0,0 – 1,0 dle převodníku analogové karty. Poté dojde ke škálování hodnoty podle měřeného rozsahu proudu na 0 – 5 A. Výsledný proud je uložen do datového bloku AI.



Obrázek 30 – Program pro převod analogových hodnot proudu

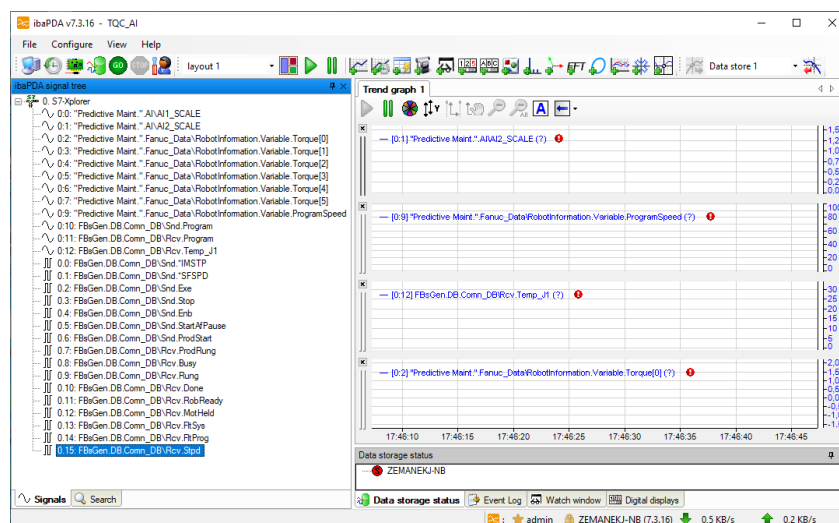
5.2 Záznam dat do souboru

Pro záznam dat je využíváno softwaru ibaPDA s licenci S7 explorer. Jedná se o software, který dokáže navázat komunikaci se Siemens PLC a zajistit záznam vybraných dat do PC. Velkou výhodou je možnost velké rychlosti vzorkování zaznamenávaných dat, která se pohybuje v jednotkách milisekund a nedochází tedy k velkým ztrátám rychlosti záznamu. Komunikace se Siemens PLC probíhá za pomoci S7 komunikace a umožňuje čtení hodnot datových bloků z programu PLC. ibaPDA nabízí nástroj pro vytvoření knihovny proměnných na základě PLC projektu. Lze tedy vybrat projekt, ze kterého mají být data čtena a dojde k vytvoření knihovny dostupných proměnných. ibaPDA nedokáže číst všechny proměnné programu a má jisté omezení. Čtená data musí být uložena v datovém bloku, u kterého musí být vypnuta optimalizace data accessu. To je z důvodu, že ibaPDA čte data z přesné adresní indexace v datovém poli. Druhou podmínkou, aby mohla být data čtena, je aktivace PUT/GET komunikace v PLC, které ibaPDA využívá. Samozřejmostí je umístění PLC a PC do jedné sítě.



Obrázek 31 – Definice S7-Explorer rozhraní v ibaPDA

Po spuštění záznamu dat lze aktuální hodnoty signálů sledovat v grafech. Proměnné, které byly definovány pro záznam, jsou zobrazeny v levém sloupci a jsou děleny do kategorií podle jejich zdroje. Při nutnosti členění lze proměnné uživatelem řadit do skupin podle vlastního uvážení.



Obrázek 32 – Sledování aktuálního průběhu zaznamenávání dat

Name	Unit	Gain	Offset	S7 Symbol	S7 Operand	S7 DataType	Active
"Predictive Maint.",AI1AI1_SCALE		1	0	BLOCKS\Blocks."Predictive Maint.",AI1AI1_SCALE	DB 31.DBD 2	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",AI1AI2_SCALE		1	0	BLOCKS\Blocks."Predictive Maint.",AI1AI2_SCALE	DB 31.DBD 12	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[0]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[0]	DB 25.DBD 0	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[1]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[1]	DB 25.DBD 4	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[2]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[2]	DB 25.DBD 8	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[3]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[3]	DB 25.DBD 12	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[4]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[4]	DB 25.DBD 16	REAL	<input checked="" type="checkbox"/>
"Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[5]		1	0	BLOCKS\Blocks."Predictive Maint.",Fanuc_Data\RobotInformation.Variable.Torque[5]	DB 25.DBD 20	REAL	<input checked="" type="checkbox"/>

Obrázek 33 – Definice zaznamenávaných proměnných v ibaPDA

Rychlost čtení dat z PLC je nastaveno na 10 ms, aby došlo k co nejrychlejšímu záznamu dat. Tento čas se může na základě vytíženosti PLC nebo komunikace měnit, ale obecně platí snaha o co nejrychlejší záznam dat. Data jsou zaznamenávána do binárního souboru s koncovkou .dat. Výhodou binárního ukládání je zachování malé velikosti souboru i při záznamu velkého množství dat. Data jsou ukládány do zvolené složky v intervalech, které jsou definovány za pomoci nastavení. Pro zachycení celého průběhu měření je definovaný počáteční signál, pomocí kterého dochází k aktivaci měření. Časový interval jednoho souboru měření je stanoven na půl hodiny. Výsledkem záznamu dat jsou .dat soubory, které odpovídají jednotlivým časovým intervalům.

Proměnné pro záznam jsou všechny čteny z datových bloků řídicího PLC robotického pracoviště. Pro účely zadání čtených proměnných byl do ibaPDA nahrán PLC program a vygenerován seznam dostupných tagů ke čtení. Z tohoto seznamu byly proměnné vybírány a přiřazeny do tabulek pro čtení analogových a digitálních hodnot. Důležitým prvkem pro čtení je absolutní adresace prvku v PLC programu, pomocí které dochází ke čtení hodnoty z paměti PLC.

Tabulka 17 – Seznam zaznamenávaných digitálních proměnných z PLC za pomoci ibaPDA

Proměnná	Cesta v PLC	Popis
FBsGen.DB.Comn_DB\Snd.*IMSTP	DB 100.DBX 28.0	Ovládání z PLC
FBsGen.DB.Comn_DB\Snd.*SFSPD	DB 100.DBX 28.1	Ovládání z PLC
FBsGen.DB.Comn_DB\Snd.Exe	DB 100.DBX 29.0	Spuštění programu
FBsGen.DB.Comn_DB\Snd.Stop	DB 100.DBX 29.1	Stop programu
FBsGen.DB.Comn_DB\Snd.Enb	DB 100.DBX 29.3	Zapnutí pohonů
FBsGen.DB.Comn_DB\Snd.StartAfPause	DB 100.DBX 29.5	Start po pauze
FBsGen.DB.Comn_DB\Snd.ProdStart	DB 100.DBX 29.6	Start produkce
FBsGen.DB.Comn_DB\Rcv.ProdRung	DB 100.DBX 0.2	Produkce běží
FBsGen.DB.Comn_DB\Rcv.Busy	DB 100.DBX 0.6	V běhu
FBsGen.DB.Comn_DB\Rcv.Rung	DB 100.DBX 1.0	Program běží
FBsGen.DB.Comn_DB\Rcv.Done	DB 100.DBX 1.1	Program dokončen
FBsGen.DB.Comn_DB\Rcv.RobReady	DB 100.DBX 1.6	Robot připraven
FBsGen.DB.Comn_DB\Rcv.MotHeld	DB 100.DBX 1.7	Pozastavený pohyb
FBsGen.DB.Comn_DB\Rcv.FltSys	DB 100.DBX 0.0	Systémová chyba
FBsGen.DB.Comn_DB\Rcv.FltProg	DB 100.DBX 0.1	Programová chyba
FBsGen.DB.Comn_DB\Rcv.Stpd	DB 100.DBX 1.2	Robot zastaven

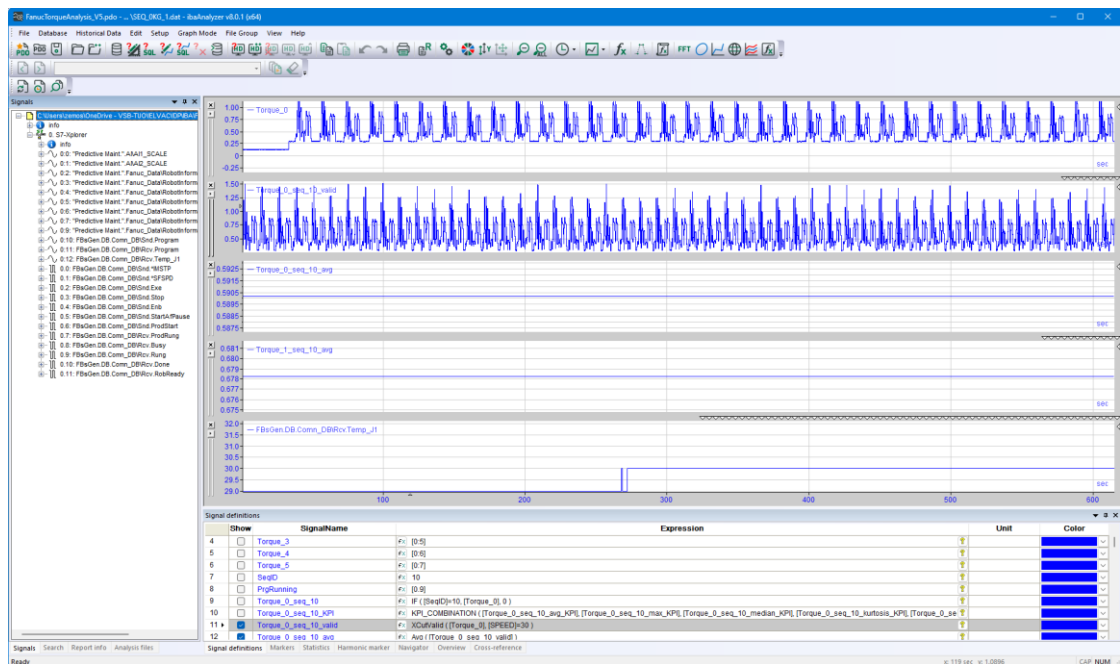
Tabulka 18 – Seznam zaznamenávaných analogových proměnných z PLC za pomoci ibaPDA

Proměnná	Datový typ	Cesta v PLC	Popis
"Predictive Maint.".AI\AI2_SCALE	REAL	DB31.DBD12	Měřená hodnota proudu na 1. ose robotu (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[0]	REAL	DB25.DBD0	Hodnota proudu 1. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[1]	REAL	DB25.DBD4	Hodnota proudu 2. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[2]	REAL	DB25.DBD8	Hodnota proudu 3. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[3]	REAL	DB25.DBD12	Hodnota proudu 4. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[4]	REAL	DB25.DBD16	Hodnota proudu 5. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.Torque[5]	REAL	DB25.DBD20	Hodnota proudu 6. osy robotu z OPC UA komunikace (A)
"PredictiveMaint.".Fanuc_Data\Robot Information.Variable.ProgramSpeed	USINT	DB25.DBB24	Nastavená rychlost robotu (%)
FBsGen.DB.Comn_DB\Snd.Program	WORD	DB100.DBW 32	Program odesílaný do robotu z PLC
FBsGen.DB.Comn_DB\Rcv.Program	WORD	DB100.DBW 4	Program přijímaný z robotu od PLC
FBsGen.DB.Comn_DB\Rcv.Temp_J1	WORD	DB100.DBW 12	Teplota 1. osy robotu (%)

Na seznamu lze najít řadu proměnných, které nejsou při analýze jinak využívány. Zaznamenávají jsou ale z důvodu možné diagnostiky poruchy, aby bylo možné zjistit v jakém stavu byl aktuálně robot. Data je také možné použít pro rozšíření systému sledování stavu robotu o sledování aktuálního chování robotu nebo aktuálně prováděného programu. Možností pro rozšíření sledování systému je mnoho a sbíraná data mohou být později využita k dalším analýzám a zpracování.

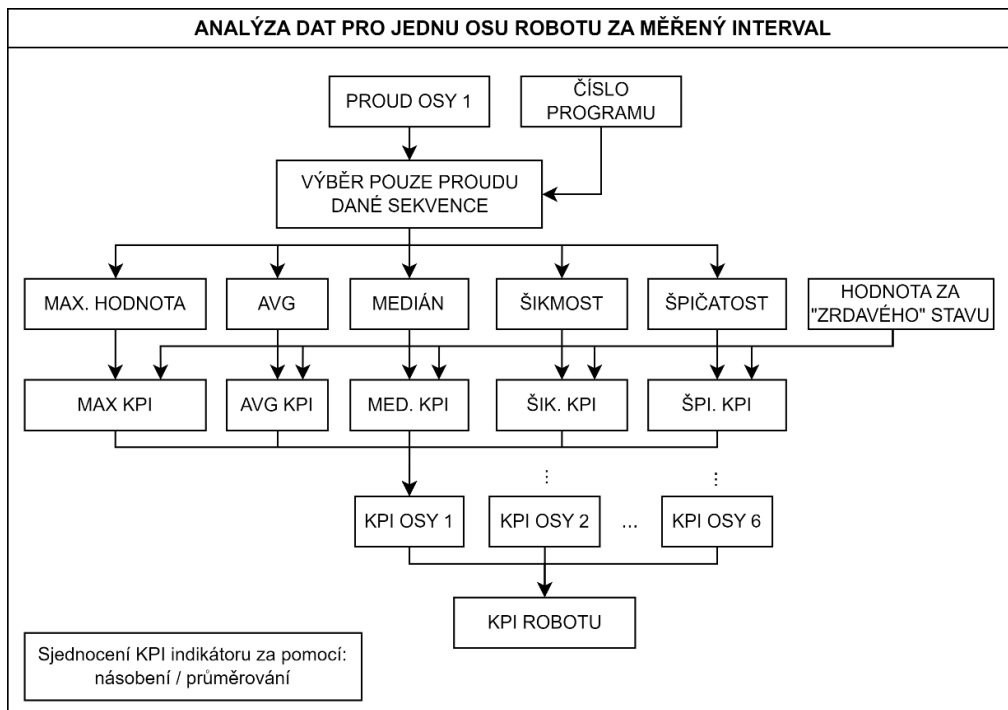
5.3 Zpracování dat

Pro zpracování dat je využit software ibaAnalyzer umožňující provedení analýzy zaznamenaných dat a jejich další zpracování. Vstupem pro ibaAnalyzer jsou .dat soubory, které byly zaznamenány za pomoci ibaPDA. Nástroj nabízí jejich zobrazení v grafech a použití pro další výpočty a práci s daty. Prostředí obsahuje velké množství nástrojů, které jsou pro práci se signály potřeba a poskytuje jednotné rozhraní pro definici analýzy dat. Součástí jsou markery, statistiky signálů, makra, vlastní funkce, možnosti nahrávání dat na databázové služby, možnost provádění SQL dotazu pro získání dat ze serveru a další užitečné nástroje. Data jsou v analýze rozdělena do jednotlivých signálů, které reprezentují danou proměnnou.



Obrázek 34 – Prostředí ibaAnalyzer

Prostředí je rozděleno do tří základních částí. V levém sloupci jsou zobrazeny načtené signály z .dat souborů. V dolní části dochází k definici vlastních signálů a psaní logiky pro zpracování signálů. Definované signály jsou poté zobrazeny v hlavním oknu programu. Na základě připravených signálů dochází k provedení analýzy zaznamenaných dat. Cílem analýzy je vyhodnocení celkového KPI robotu a proces zpracování dat je blokově popsán na následujícím obrázku (Obrázek 35).



Obrázek 35 – Analýza – Schéma analýzy dat pro jednu osu robotu

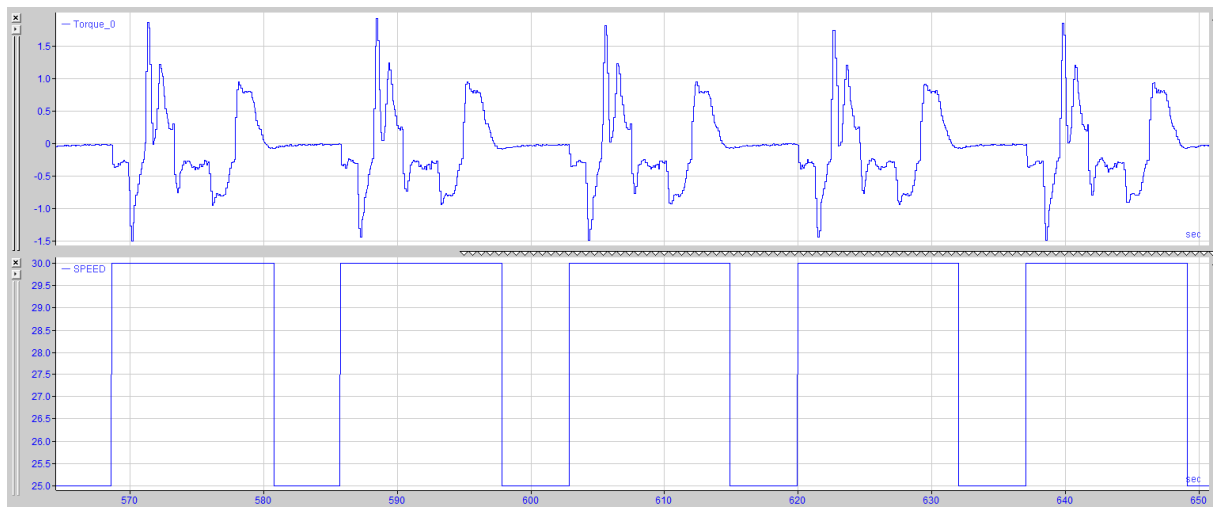
Na samotném počátku analýzy dojde k otevření připraveného souboru analýzy ve formátu .pdo. Jedná se o formát analýzy ibaAnalyzer, který v sobě ukládá veškeré operace a výpočty, které jsou pro dané

signály připravené. Po otevření analýzy dochází k načtení datového souboru .dat, který obsahuje zaznamenaná data z PLC za daný časový úsek. Soubor analýzy okamžitě načte potřebné signály na základě jejich ID a provede výpočet navazujících signálů. Výsledky analýzy jsou poté připraveny pro uložení na databázový server. Postupu exportu do DB se věnuje další kapitola.

V následující části je popsán proces výpočtu KPI pro jednu osu robotu (jeden analyzovaný průběh proudu) a následné sjednocení do jednoho hlavního KPI indikátoru reprezentující výsledný stav robotu. Na začátku dojde k definici signálů z jednotlivých zaznamenaných dat robotu. Definují se signály o průběhu proudu na ose, číslo programu robotu, rychlosti robotu a proměnné o stavu vykonávání programu (Rung). Jelikož je pro testovací sekvenci spuštěn pouze jeden program, vliv čísla programu lze zanedbat. V případě analýzy více sekvencí (programů) by došlo za pomoci čísla programu k rozdělení na více analýz. Analyzovaný průběh lze z celkového signálu rozpoznat na základě proměnné Rung, která signalizuje běh zvoleného programu. Jelikož se proměnná Rung čte za pomoci Modbus komunikace, dochází u proměnné k poměrně značnému zpoždění, které může mít vliv na výsledné výpočty. Aby byl analyzovaný průběh definován přesněji, je provedena selekce na základě detekce rychlosti robotu, která je za tímto účelem při běhu programu navýšena.

Nejprve dojde k načtení signálu „Torque_0“ (A) reprezentující proud na 1. ose robotu a rychlosti robotu SPEED (%) pro rozpoznání analyzovaného intervalu.

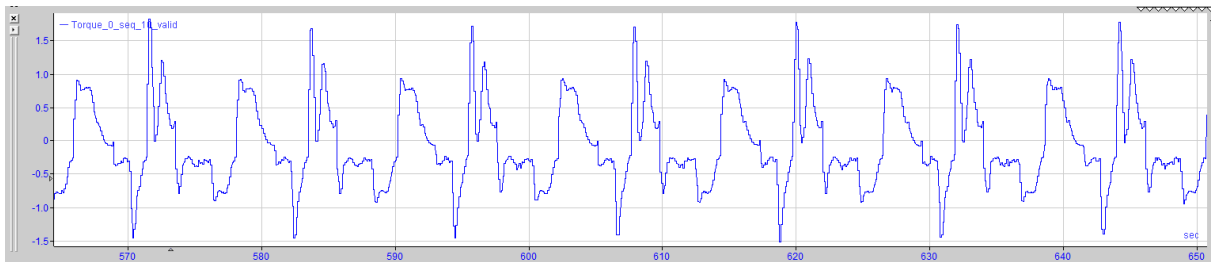
```
//Definice signálů  
Torque_0 = [0:2]  
SPEED = [0:9]
```



Obrázek 36 – Analýza – Načtení signálu proudu a rychlosti

Aby výpočet statistických hodnocení nebyl ovlivněn dobou, po kterou není vykonávaný program a robot pouze čeká na zavolání programu, je nutné tyto oblasti signálu vyjmout. K tomu slouží právě signál „SPEED“, který při hodnotě 30 detekuje spuštěný program a při hodnotě 25 dochází k čekání robotu na spuštění programu. Za pomoci následující definice signálu dojde k zanechání pouze průběhu, kdy je hodnota rovna hodnotě 30 a spojení analyzovaných průběhů k sobě.

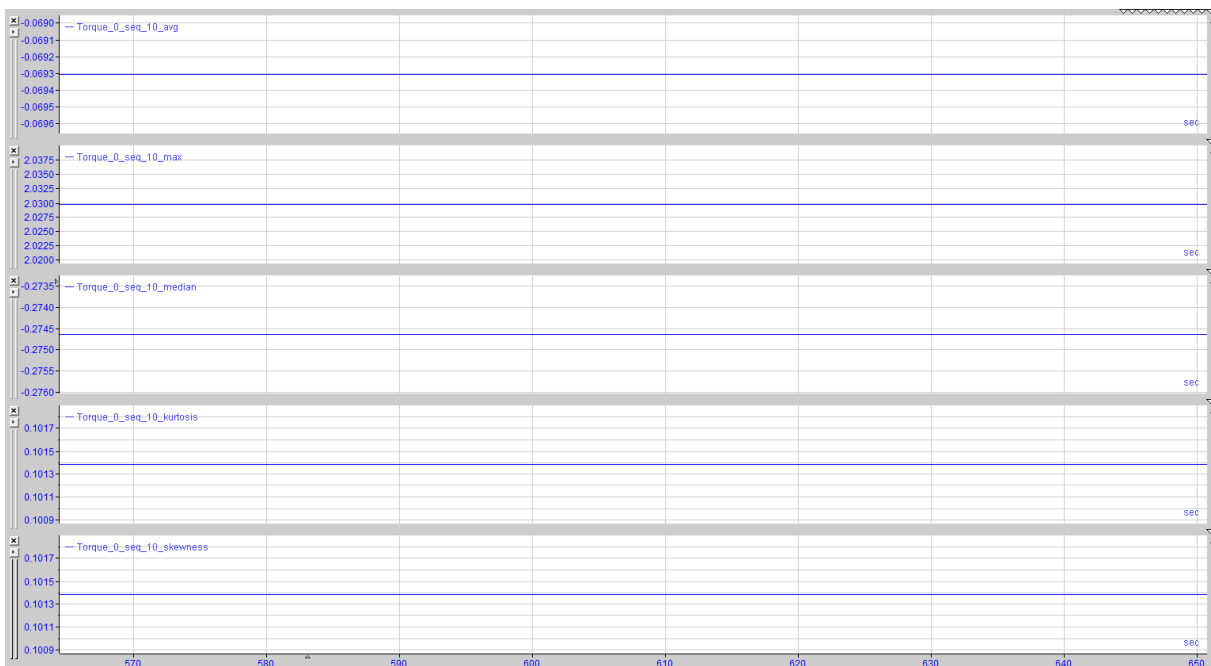
```
//Definice signálů  
Torque_0_valid = XCutValid ( [Torque_0], [SPEED]=30 )
```



Obrázek 37 – Analýza – Selekce analyzovaných průběhů

Takto připravený průběh již lze jako celek hodnotit navrženými algoritmy. Z připraveného signálu dochází k výpočtu pěti statistických hodnocení, které mají za úkol zhodnotit naměřený průběh signálu a stanovit parametry vhodné pro porovnávání s průběhy v jiných časech a zatížení robotu. Statistické hodnocení průběhu jsou počítány na základě již existujících funkcí v ibaAnalyzeru.

```
//Definice signálů
Torque_0_avg = Avg ( [Torque_0_valid] )
Torque_0_max = Max ( [Torque_0_valid] )
Torque_0_median = Median ( [Torque_0_valid] )
Torque_0_kurtosis = Kurtosis ( [Torque_0_valid] )
Torque_0_skewness = Skewness ( [Torque_0_valid] )
```



Obrázek 38 – Analýza – Výpočet statistických hodnocení průběhu

Na základě vypočtených statistických hodnocení dochází k výpočtu jejich KPI hodnot. Hodnota KPI se počítá z vypočtené hodnoty statistických hodnocení a „zdravé“ hodnoty stanovené na začátku provádění prediktivní údržby, se kterou se aktuální hodnota porovnává. Výsledkem výpočtu je KPI hodnota od 0 až 100 %, která reprezentuje odchylku od „zdravé“ hodnoty. Během výpočtu je nutné ošetření porovnávání hodnot s jinými znaménky, aby docházelo ke správnému výpočtu KPI. Výpočet se provádí opakovaně a pro zjednodušení zadávání výpočtu KPI je výpočet definován jako makro. Vstupními parametry makra je aktuální hodnota a „zdravá“ hodnota. Výstupem je vypočtená KPI hodnota popisující odchylku aktuální hodnoty od „zdravé“. Za účelem ošetření bylo při výpočtu

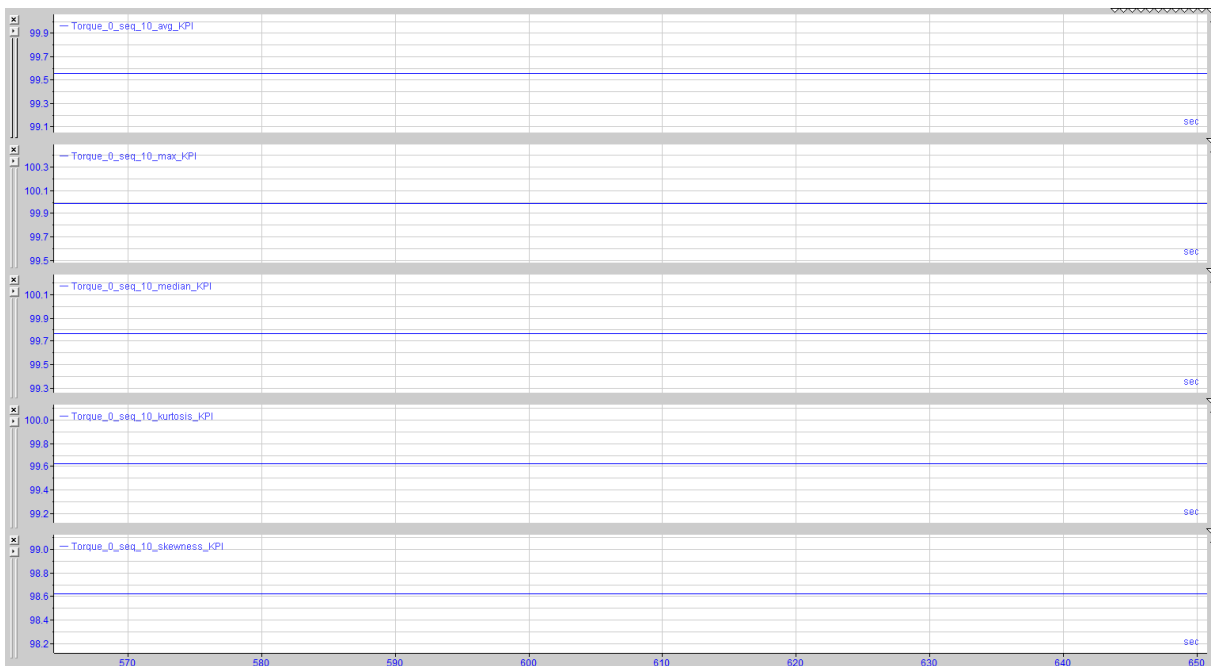
počítáno s porovnáním hodnot rozdílných polarit a velikostí. Došlo také k ošetření, aby vždy byla maximální hodnota KPI rovna 100 %.

```
//Definice makra KPI(ideal_value,actucal_value)
//Vstupy
ideal_value
actucal_value

//vnitřní proměnné
KPI_RAW = IF ( ( [ideal_value] < 0 AND [actucal_value] > 0 ) OR ( [ideal_value] > 0 AND
[actucal_value] < 0 ), [actucal_value] + 20 / ( [ideal_value] + 20 / 100 ), [actucal_value] /
( [ideal_value] / 100 ) )
KPI_SIDE_FIX = IF ( [KPI_RAW] > 100, 100 - ( [KPI_RAW] - 100 ), [KPI_RAW] )

//Výstup
KPI_OUTPUT = IF ( [KPI_SIDE_FIX]<0, 0, [KPI_SIDE_FIX] )
```

```
//Definice signálů
Torque_0_avg_KPI = KPI ( -0.069, [Torque_0_avg] )
Torque_0_max_KPI = KPI ( 2.03, [Torque_0_max] )
Torque_0_median_KPI = KPI ( 2.03, [Torque_0_median] )
Torque_0_kurtosis_KPI = KPI ( 0.1010, [Torque_0_kurtosis] )
Torque_0_skewness_KPI = KPI ( 0.1, [Torque_0_skewness] )
```



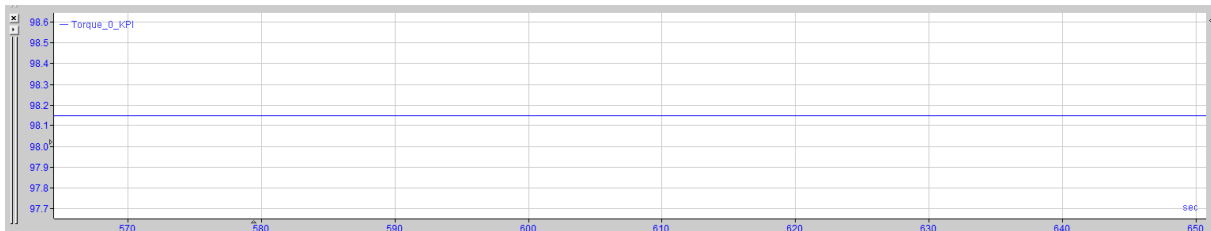
Obrázek 39 – Analýza – výpočet KPI statistických hodnotení

Takto jsou vypočteny KPI hodnoty pro jednotlivé statistické veličiny a následně jsou kombinovány do jednoho celkového KPI pro hodnocení stavu jedné osy robotu. K tomuto kroku je opět definováno makro pro jednodušší zadávání kombinovaných hodnot.

```
//Definice signálů
Torque_0_KPI = KPI_COMBINATION ( [Torque_0_avg_KPI], [Torque_0_max_KPI],
[Torque_0_median_KPI], [Torque_0_kurtosis_KPI], [Torque_0_skewness_KPI] )
```

```
//Definice makra KPI_COMBINATION(KPI1,KPI2,KPI3,KPI4,KPI5)
//Vstupy
KPI1
KPI2
KPI3
KPI4
KPI5

//Výstup
KPI_COMB = [KPI1] / 100 * [KPI2] / 100 * [KPI3] / 100 * [KPI4] / 100 * 100
```



Obrázek 40 – Analýza – výpočet KPI osy robotu

Po obdobném výpočtu KPI pro všechny osy robotu, jsou stejnou funkcí KPI(comb) zkombinovány do jednoho KPI hodnotícího stav celého robotu.

```
//Definice signálů
Robot_seq_10_KPI = [Torque_0_KPI] / 100 * [Torque_1_KPI] / 100 * [Torque_2_KPI] / 100 *
[Torque_3_KPI] / 100 * [Torque_4_KPI] / 100 * [Torque_5_KPI] / 100 * 100
```

5.4 Uložení výsledků analýzy na SQL server

IbaAnalyzer nabízí dva možné způsoby exportování výsledků provedené analýzy a uložení měřených nebo vypočtených signálů. Ve výchozí konfiguraci je zvolen export do datového souboru .dat, který zvolený výstup analýzy uloží do jednoho souboru. Při exportu je možné zvolit typ přípony souboru a to na .dat, .csv nebo některých dalších typů. Z důvodu často vysokého objemu ukládaných dat je výhodné použití binárního formátu .dat, který například oproti .csv výrazně šetří místo na disku. Tato funkcionality je součástí IbaAnalyzer a nevyžaduje dodatečné licence.

Druhou nabízenou možností je export výstupních dat analýzy do databáze, která byla využita při této práci. Po provedeném zpracování jsou data nahrány na SQL server, který zajišťuje jejich archivaci a dlouhodobý sběr. Jistou variantou bylo uchovávání dat ve formátu .dat, ale pro další práci s daty není vhodný. Funkcionality nahrávání výsledků na databázový server je nutno licencovat licenci IbaAnalyzer-DB, která zpřístupňuje funkce DB exportu, SQL dotazu a další funkce spojené s kooperací s databázovým propojením. Jakožto databázový server byl zvolen MS SQLEXPRESS.

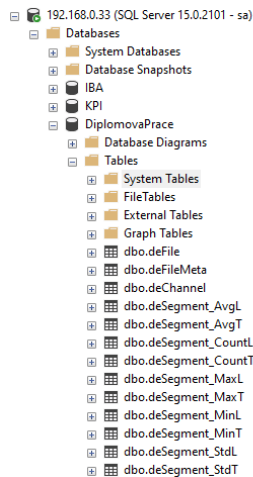
Name	State	Start Mode	Log On As	Process ID	Service Type
SQL Server (SQLEXPRESS)	Running	Automatic	NT Service\MSSQLS...	6624	SQL Server
SQL Server (MSSQLSERVER)	Running	Automatic	LocalSystem	6316	SQL Server

Obrázek 41 – Běžící SQL Express server

Databázový server běží lokálně na stejném počítači, kde dochází k záznamu a analýze dat. Databázi lze však provozovat na kterémkoliv počítači v síti a není nutno umístění na stejný stroj. Na serveru došlo k vytvoření databáze „IBA“, která slouží jako primární databáze pro archivaci dat. Jednotlivé tabulky byly již automaticky vytvořeny programem IbaAnalyzer, který obsahuje vlastní strukturu dělení dat do tabulek.

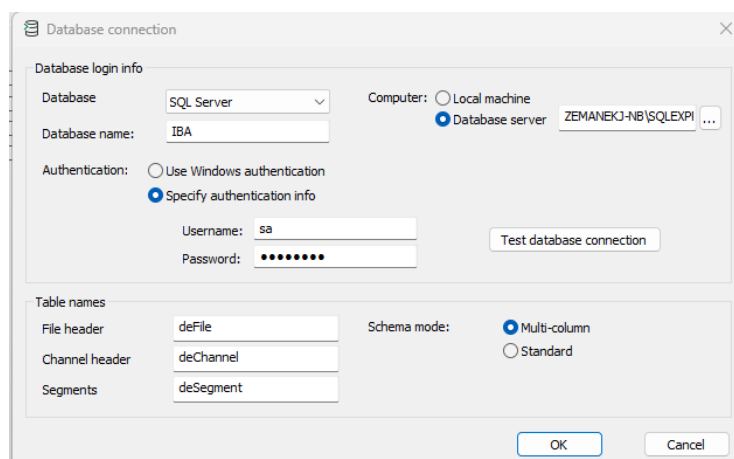
Struktura hlavních tabulek ibaAnalyzer-DB:

- **deFile** – Uložení informací o analyzovaném .dat souboru, uložení timestamp údaje, kdy byla zaznamenána první hodnota .dat souboru
- **deSegment_AvgT**– Uložení hodnot signálů společně s číslem segmentu, který definuje posun hodnoty v čase od první hodnoty v závislosti na vzorkovací frekvenci (+1 segment = jeden krok vzorkování signálu)



Obrázek 42 – Automaticky generována struktura tabulek pro ukládání dat

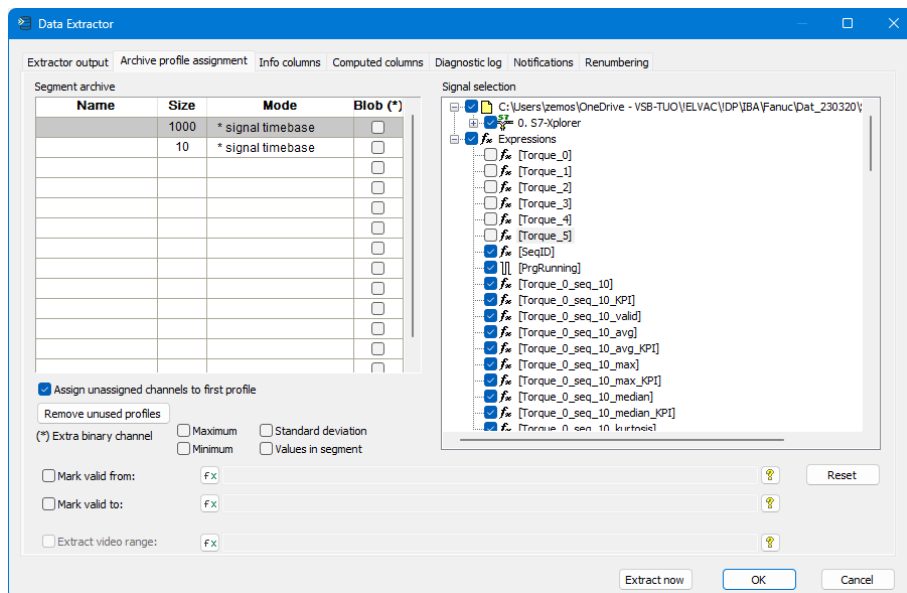
V dialogovém okně ibaAnalyzeru došlo k definici připojení k databázovému serveru a vyplnění informací o konkrétní databázi a tabulkách, do kterých budou data nahrávána. Velmi důležitým prvkem je zvolení Multi-column struktury ukládání dat. Ta zaručuje ukládání hodnot jednoho signálu do jednoho sloupce v závislosti na timestamp údaji v řádcích. Toto specifické formátování je nutné pro snadné vyčtení hodnot do vizualizace dat. Pokud by došlo k nahrávání dat do databáze ve formátu standard, došlo by k rozdělení dat do segmentů a další vyčítání by bylo téměř nemožné.



Obrázek 43 – Definice připojení databáze

Pokud je správně nastavené připojení k databázi, je možné zahájit extrakci dat do databáze. Extrakce samotná obsahuje spoustu nastavení a možností, které definují, co a jak se do databáze nahráje. Nejdůležitějším nastavením extrakce je volba signálů a s jakým časovým intervalem budou do databáze nahrávány. Definici zobrazuje následující okno nastavení (Obrázek 44). V levé části se definuje vzorkovací frekvence, s kterou jsou signály do databáze nahrávány a v pravé části se přiřazují signály

k dané vzorkovací frekvenci. V práci jsou použity dvě vzorkovací frekvence, pomalá 1000násobek vzorkování původního signálu (vzorkování co 10 s výsledného signálu) a rychlá 10násobek (vzorkování co 0,1 s výsledného signálu). Pomocí rychlé vzorkovací frekvence jsou do databáze ukládány zdrojové signály proudu, aby byla ve vizualizaci možná jejich rekonstrukce bez ztráty špiček v průběhu. Pomalá frekvence je použita pro vypočtená data, která jsou během celého měřeného intervalu stejná a není důvod jejich rychlé vzorkování. Při dlouhodobém sběru dat by bylo vhodné zvolit ještě menší vzorkování, a to z důvodu ušetření místa v databázi a rychlejšího čtení hodnot do vizualizace.



Obrázek 44 – Definice nahrávání signálů do databáze

Pro export dat do databáze lze stisknout tlačítko „extract now“, které extrakci provede. Tento manuální postup je možný a během práce byl využíván především pro testování. Pro zpracování více dat byl používán automatický systém extrakce, který je popsán v jedné z následujících kapitol. Výsledek extrakce do databáze za použití multicolumn formátu je vidět na následujícím obrázku.

Field	SegmentNr	Torque_0	Torque_1	Torque_2	Torque_3	Torque_4	Torque_5	SeqID	PrgRunning	Torque_0_seq_10	Torque_0_seq_10_KPI	Torque_0_seq_10_valid	Torque_0_seq_10_avg	Torque_0_seq_10_avg_KPI	Torque_0_seq_10_max	Torque_0_seq_10_max_KPI	
1	829062495	0	-0.3188684	0.5569898	1.76078	-0.073332	0.1347157	0.8720132	10	1	-0.3188684	0	-0.3188684	-0.006181452	95.23192	1.307333	99.
2	829062495	1	-0.2903049	0.7892341	1.813787	-2.471299	0.04916232	0.7959352	10	1	-0.2903049	0	-0.2903049	-0.006181452	95.23192	1.307333	99.
3	829062495	2	-0.2889316	0.8223016	1.826421	-2.29772	0.03515518	0.7973084	10	1	-0.2889316	0	-0.2889316	-0.006181452	95.23192	1.307333	99.
4	829062495	3	-0.2452623	0.9518516	1.805273	-1.859709	-0.01180594	0.7452623	10	1	-0.2452623	0	-0.2452623	-0.006181452	95.23192	1.307333	99.
5	829062495	4	-0.2809668	1.033233	1.765386	-1.318044	-0.01891211	0.7600934	10	1	-0.2809668	0	-0.2809668	-0.006181452	95.23192	1.307333	99.
6	829062495	5	-0.2592694	1.055479	1.791815	-1.149684	0.00755287	0.7530998	10	1	-0.2592694	0	-0.2592694	-0.006181452	95.23192	1.307333	99.
7	829062495	6	-0.2466355	1.073331	1.73304	-1.034606	0.002609173	0.7105191	10	1	-0.2466355	0	-0.2466355	-0.006181452	95.23192	1.307333	99.
8	829062495	7	0.6575117	1.044592	1.795111	-1.252678	0.07127163	-0.4391651	10	1	0.6575117	0	0.6575117	-0.006181452	95.23192	1.307333	99.
9	829062495	8	1.10217	1.041747	1.861027	-1.455369	0.1663005	-0.972535	10	1	1.10217	0	1.10217	-0.006181452	95.23192	1.307333	99.
10	829062495	9	0.9431475	1.039	1.814886	-1.49437	0.1752266	-1.150783	10	1	0.9431475	0	0.9431475	-0.006181452	95.23192	1.307333	99.
11	829062495	10	0.581159	1.031585	1.78028	-1.355672	0.119198	-1.23153	10	1	0.581159	0	0.581159	-0.006181452	95.23192	1.307333	99.
12	829062495	11	0.2287833	1.053282	1.737435	-1.110135	0.1180994	-1.181269	10	1	0.2287833	0	0.2287833	-0.006181452	95.23192	1.307333	99.
13	829062495	12	0.1746773	1.042571	1.732491	-1.014007	0.1403461	-1.202956	10	1	0.1746773	0	0.1746773	-0.006181452	95.23192	1.307333	99.
14	829062495	13	0.2425158	1.072508	1.763252	-1.032683	0.1260643	-1.150434	10	1	0.2425158	0	0.2425158	-0.006181452	95.23192	1.307333	99.
15	829062495	14	0.2686075	1.121121	1.78028	-0.9810492	0.1183741	-1.1961	10	1	0.2686075	0	0.2686075	-0.006181452	95.23192	1.307333	99.
16	829062495	15	0.3523757	2.012908	1.045592	1.344686	0.09420489	0.4270805	10	1	0.3523757	0	0.3523757	-0.006181452	95.23192	1.307333	99.
17	829062495	16	0.3699533	2.253227	0.747871	2.137599	0.08349355	0.7659688	10	1	0.3699533	0	0.3699533	-0.006181452	95.23192	1.307333	99.
18	829062495	17	0.4086789	2.522384	0.799231	3.474869	0.1612194	1.241966	10	1	0.4086789	0	0.4086789	-0.006181452	95.23192	1.307333	99.

Obrázek 45 – Ukázka tabulky deSegment_AvgT s nahranými signály

5.5 Vizualizace dat (Grafana)

Pro účely zobrazení uložených dat z databáze je zvolena webová aplikace Grafana. Jedná se o open source platformu, která nabízí vizualizaci dat v grafech, tabulkách a dalších prvcích, které jsou shlukovány do tzv. dashboardů (obrazovek). Podporuje rozšiřující pluginy, data sources, aletry a další funkce. Pro účely práce je Grafana používána lokálně ve verzi „Self-managed“, která nepodléhá žádnému licencování či předplatnému. Jak je zmíněno v předchozím bodě, zdrojem dat pro vizualizaci

je MS SQL databáze. Pro načtení dat z databáze je využíváno SQL dotazů, které jsou definovány pro každý panel vizualizace konkrétně na základě požadovaných signálů.

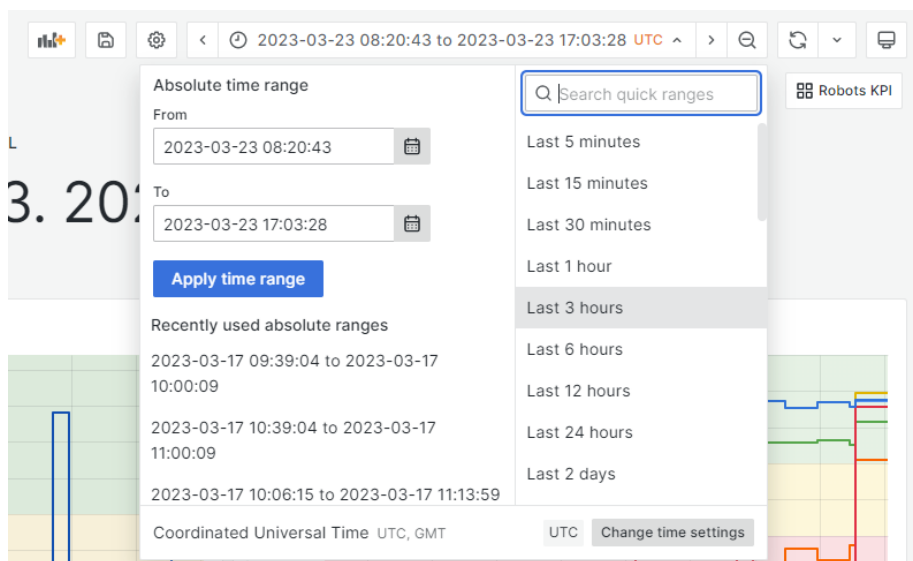
Typy panelů

Grafana nabízí širokou nabídku panelů, které jsou slučovány do obrazovek a tvoří tím samotnou vizualizaci. Každý panel nabízí určitou funkčnost nebo jistou formu vizualizace dat. V následujících odřádkách jsou popsány prvky použité v této práci. Zdrojem všech dat v panelech je SQL dotaz.

- **Time chart** – Časový průběh sledované veličiny
- **Gauge** – Vizualizace jedné hodnoty, nebo hodnoty v rámci intervalu. Integrace barevného stavu.
- **Histogram** – Vizualizace rozdělení proměnné v intervalech
- **Text** – zobrazení textu nebo proměnných v rámci dashboardu
- **Alarms** – zobrazení aktivních alarmových hlášení
- **Table** – tabulka zobrazující údaje z databáze
- **State timeline** – zobrazení stavu v čase na základě definovaných podmínek

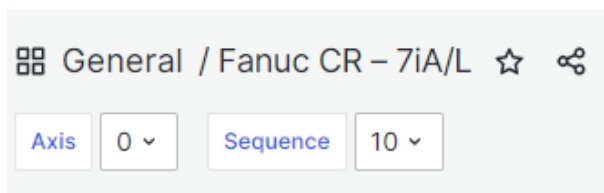
Volba časového období zobrazených dat

Veškerá data zobrazována v panelech jsou vázána na časový údaj (timestamp). Volba časového intervalu hodnot se provádí v pravém horním rohu za pomoci definice konkrétního intervalu nebo dynamického zobrazování vzhledem k aktuálnímu času. Součástí panelu je možnost zapnutí automatické obnovy dat a intervalu obnovy.



Obrázek 46 – Vizualizace – volba zobrazeného časového období

Aby bylo umožněno uživatelem vybírat, která data chce zobrazit, nabízí Grafana integraci proměnných, které lze používat v SQL dotazech. V práci je použita především proměnná $\$Axis$, pomocí které uživatel vybírá osu, ke které chce zobrazit data a grafy. Tato proměnná je vepsána přímo do SQL dotazu, takže po změně proměnné dojde k obnově dat ze strany DB a zobrazení dat jiné osy robotu. Druhou využívanou proměnnou je proměnná $\$Sequence$, která vybírá analyzovanou sekvenci robotu. Během návrhu a testování systému byla využívána, ale pro účely finální prezentace systému prediktivní údržby došlo k zaměření pouze na jednu sekvenci číslo 10. Jedná se o vlastní proměnné, tvořené pro účely konkrétního použití.



Obrázek 47 – Vizualizace – Volba uživatelských proměnných

V Grafaně existují také systémové proměnné, které jsou již definovány a poskytují globální informace o vizualizaci. Příkladem takových proměnných, které jsou v práci využity, jsou proměnné `$_from` a `$_to`. Jedná se o proměnné, které obsahují údaje o zvoleném časovém období a lze je použít do SQL dotazu pro získání správných dat. Grafana obsahuje spoustu dalších vnitřních proměnných, které lze použít na vytažení systémových informací a použití ve vlastním SQL dotazu nebo kódu.

Obdobně jako pro přepínání hodnot proměnných, nabízí Grafana také prvky pro přepínání dashboardů. Prvky jsou umístěny v pravém horním rohu pod volbou časového intervalu. Po přidání požadovaných dashboardů lze jednoduše za pomoci jednoho kliknutí dashboard přepnout a načíst jiný dashboard. Nevýhodou je absence možnosti úpravy velikosti nebo umístění tlačítek. Naopak výhodou je možnost udržení stejného časového intervalu napříč dashboardy, takže nedojde ke ztrátě pozorovaného časového období. V rámci nastavených pravomocí přihlášeného uživatele lze nastavit, kdo má ke kterým dashboardům přístup a co může provádět za změny. Lze definovat také uživatele, který má zobrazený jeden dashboard a nemá možnost žádného přepnutí.

Úprava časového údaje mezi databází a vizualizací

Data nejsou v databázi ukládány pod jednotným timestampem a pro účely vizualizace je nutná jejich úprava do správného formátu. V databázi jsou data pro výpočet timestampu rozdělena do dvou tabulek.

Tabulka 19 – Rozdělení časových značek v databázi

Tabulka	Sloupec	Data ve sloupci
deFile	_TimeStamp	2022-12-14 17:04:40
	_FileId	1603138751
deSegment_AvgT	_SegmentNr	1,2,3,4,5 ... x
	_FileId	1603138751

Jednotlivá data dané proměnné jsou v tabulce deSegment_AvgT spjata pouze s číslem segmentů, při čtení by tedy byla data pouze řazena za sebe bez odpovídajícího časového údaje. Grafana takový průběh ani nevykreslí, jelikož požaduje celý časový údaj daných proměnných. Časový údaj je nutné dopočítat za pomoci sloupce _TimeStamp v tabulce deFile, kde dochází k uložení prvního časového údaje zpracovávaného .dat souboru. Za pomoci spojení tohoto timestampu a přičítání čísla segmentu proměnné lze dostat konkrétní čas, kdy byla proměnná zaznamenána. Pro výpočet je využito vztahu:

```
$_timeEpoch(_TimeStamp + _SegmentNr), //výpočet timestampu bez úpravy vzorkování
FROM
  deSegment_AvgT,deFile
WHERE deSegment_AvgT._FileId = deFile._FileId
```

Aby došlo k propojení správných dat, obsahuje SQL dotaz klauzuli WHERE, která zajišťuje počítání času ze správného deFile záznamu. Sloupec _SegmentNr určuje u každého záznamu jeho pořadí vzhledem k timestampu začátku záznamu v deFile tabulce. Pořadí hodnot je číslováno za pomoci řady celých čísel 1,2,3...atd. V údajích ale není zanesena informace o vzorkovací frekvenci, s kterou byla data do databáze nahrána. Z tohoto důvodu je nutné pořadové číslo hodnoty proměnných upravit na základě aplikované vzorkovací frekvence tak, aby odpovídalo času provedení jednoho vzorku. Pro základní úpravu na jednotky sekundy je nutné _SegmentNr dělit hodnotou 24 pro převod na hodiny, hodnotou 60 pro převod na minuty a opět hodnotou 60 pro převod na sekundy.

Výpočet závisí také na nastaveném vzorkování dat do databáze (viz Obrázek 44), na kterém lze vidět vzorkování 1000násobkem a 10násobkem původní vzorkovací frekvence signálu (0,01 s). Pro převod je tedy nutná další úprava ze sekund na konkrétní vzorkovací frekvenci _SegmentNr.

Tabulka 20 – Převod vzorkování mezi databází a vizualizací

Vzorkovací frekvence signálu (s)	Vzorkování při exportu do DB	Hodnota nastaveného vzorkování	Vzorkování signálu při nahrání do DB (s)	Přepočet _SegmentNr ve Vizualizaci
0,01	* vzorkovací frekvence	1000	10	$_SegmentNr/24.0/60/60*10$
0,01	* vzorkovací frekvence	10	0,1	$_SegmentNr/24.0/60/60/10$

Příklad kompletního SQL dotazu pro vyčítání signálů do panelu trendu společně s integrací přepočtu časového údaje (vzorkování co 10 s) lze pozorovat v následujícím kódu.

```
SELECT
  $__timeEpoch(_TimeStamp + _SegmentNr/24.0/60/60*10),
  Torque_${axis}_seq_${seq}_avg_KPI as "AVG",
  Torque_${axis}_seq_${seq}_max_KPI as "MAX",
  Torque_${axis}_seq_${seq}_median_KPI as "Median",
  Torque_${axis}_seq_${seq}_kurtosis_KPI as "Šikmost",
  Torque_${axis}_seq_${seq}_skewness_KPI as "špičatost"
FROM
  deSegment_AvgT, deFile
WHERE deSegment_AvgT._FileId = deFile._FileId
```

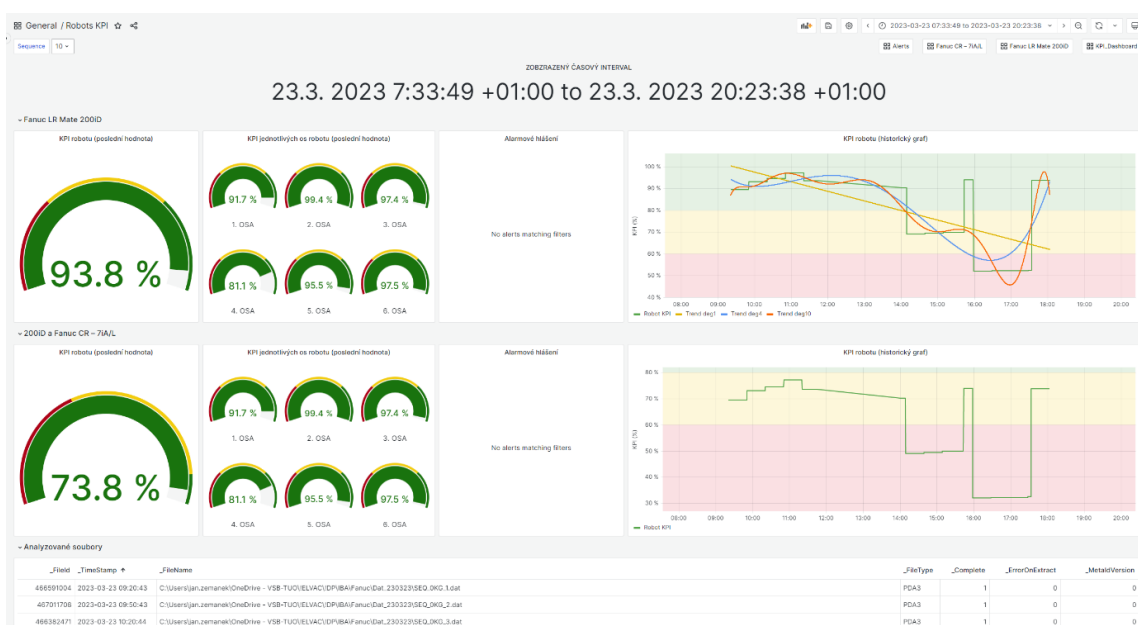
Obrazovky vizualizace

Samotná vizualizace stavu robotického pracoviště je rozdělena do dvou typů dashboardů. Prvním typem je vizualizace přehledu stavu obou robotických manipulátorů. Obrazovka obsahuje veškeré důležité prvky, které na první pohled popisují stav pracoviště a historii jeho stavu. Na dashboardu jsou zobrazeny data o obou robotech, ale jak již bylo zmíněno na začátku této kapitoly, primárním cílem je prezentace funkčnosti systému prediktivní údržby pouze na jednom robotu. Data pro druhého robota budou tedy ve vizualizaci vycházet z dat pro prvního a budou pouze upraveny konstantou.

Dashboard pro zobrazení globálního stavu robotů zobrazuje:

- **Časový interval.** Zobrazení časového intervalu pozorovaných dat.
- **Poslední hodnoty KPI indikátory os robotů.** Zobrazení posledních KPI hodnot jednotlivých os KPI robotů ve zvoleném časovém intervalu.

- **Poslední hodnoty KPI indikátory robotů.** Panely zobrazují poslední hodnotu KPI robotu ve zvoleném časovém intervalu. KPI hodnota je vyjadřována v procentech, kde 100 % je nejlepší stav a 0 % nejhorší stav.
- **Alarmové hlášení,** které zobrazuje veškeré aktivní alarmy v Grafaně. Alarmy jsou definovány pro poklesy KPI robotů pod stanovenou hodnotu.
- **Trend hodnoty KPI** v časovém intervalu, který zobrazuje historii stavu robotu v čase. Tento graf je aproximován křivkou, která pomáhá reprezentovat vývojový trend KPI. Trend KPI je opět uváděn v %.
- **Tabulky analyzovaných souborů.** Podrobnosti o datových souborech, ze kterých byla data zpracována.



Obrázek 48 – Vizualizace – obrazovka přehledu o stavu robotů

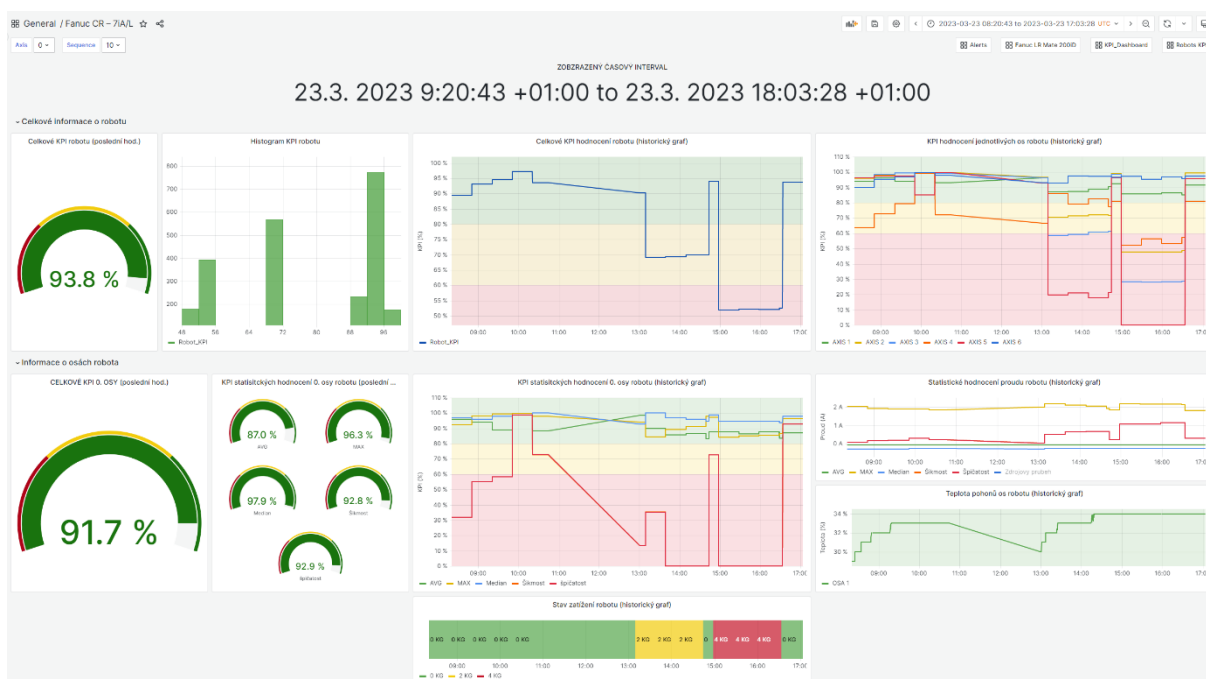
Druhý typ obrazovky zobrazuje již konkrétní údaje o robotu. Na dashboard konkrétního robotu je možné přejít z pravého horního rohu. Obrazovka zobrazuje podrobnou analýzu KPI robotu od zdrojových dat až po výsledné hodnocení robotu jako celku. Umožňuje pohled na složení KPI a na analýzy jednotlivých os. Data jsou vždy vztažena na zvolený časový interval, vybranou osu a sekvenci. Oblast pro celkové informace o robotu zobrazuje:

- **Časový interval.** Zobrazení časového intervalu pozorovaných dat.
- **Poslední hodnotu KPI indikátory robotu.** Panel zobrazuje poslední hodnotu KPI robotu ve zvoleném časovém intervalu.
- **Histogram,** zobrazující rozložení KPI hodnoty ve zvoleném časovém intervalu.
- **Trend hodnoty KPI robotu** v časovém intervalu, který zobrazuje historii stavu robotu v čase.
- **Trend hodnot KPI os robotu** v časovém intervalu, který zobrazuje historii stavu os robotu v čase.

Oblast pro informace o konkrétní ose robotu zobrazuje (volba osy za pomoci proměnné):

- **Poslední hodnotu KPI indikátoru osy robotu.** Panel zobrazuje poslední hodnotu KPI indikátoru zvolené osy robotu ve zvoleném časovém intervalu.

- **Poslední hodnoty KPI indikátorů statistických hodnocení osy robotu.** Panel zobrazuje poslední hodnoty KPI hodnocení zvolené osy robotu ve zvoleném časovém intervalu.
- **Trend hodnot KPI indikátorů statistických hodnocení osy robotu** ve zvoleném časovém intervalu.
- **Trend statistických hodnocení osy robotu** ve zvoleném časovém intervalu. Zobrazuje hodnoty statistických hodnocení osy robotu.
- **Trend teploty osy robotu** ve zvoleném časovém intervalu.
- **Stavový graf** zobrazující stav zatížení robotu v čase.



Obrázek 49 – Vizualizace – Obrazovka konkrétních informací o stavu robotu

5.6 Realizace predikce stavu

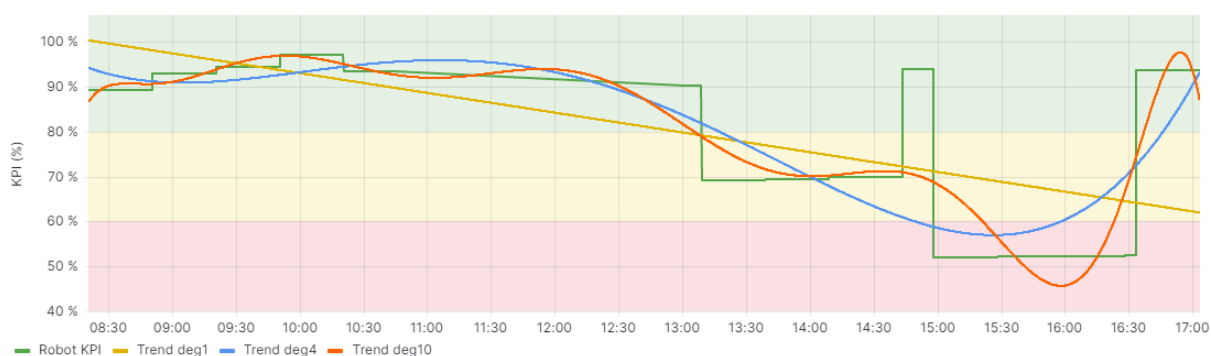
Jistou predikci stavu lze odvodit již z pohledu na KPI průběh ve vizualizaci. K lepší představě o vývoji trendu hodnoty KPI je součástí systému proložení výsledného průběhu za pomoci polynomu x -tého řádu. Proložení hodnot je realizováno za pomoci smyčky získání dat za zvolené časové období z databáze, provedení výpočtu polynomu proložení a poté opětovné nahrání vypočtené aproximace do databáze. I přes hledání možností a snahy automatizace tohoto procesu, používané nástroje iba aktuálně nenabízí možnost pro provedení automatického SQL dotazu, zpracování dat a nahrání výsledků zpět do databáze. Realizaci aproximace je tedy nutné provést ručně za pomoci připraveného souboru analýzy a SQL dotazu.

```
//SQL dotaz pro načtení průběhu
SELECT _TimeStamp + _SegmentNr/24.0/60/60/10,
Robot_seq_10_KPI as "Robot_KPI"
FROM
    deSegment_AvgT, deFile
WHERE deSegment_AvgT._FileId = deFile._FileId
ORDER BY _TimeStamp + _SegmentNr/24.0/60/60/10
```

Pro účely ukázky kódu byla zvolena aproximace 2. řádu a výpočet byl proveden za pomoci integrovaných funkcí v ibaAnalyzeru. Velikost polynomu lze ve funkcích velmi jednoduše měnit.

```
// Výpočet křivky aproximace průběhu KPI
Deg2_Coefffs = LSQPolyCoef ( XValues ( [KPI] ), [KPI], 2 )
Deg2_trend = Polynomial ( [Deg2_Coefffs], XValues ( [KPI] ) )
```

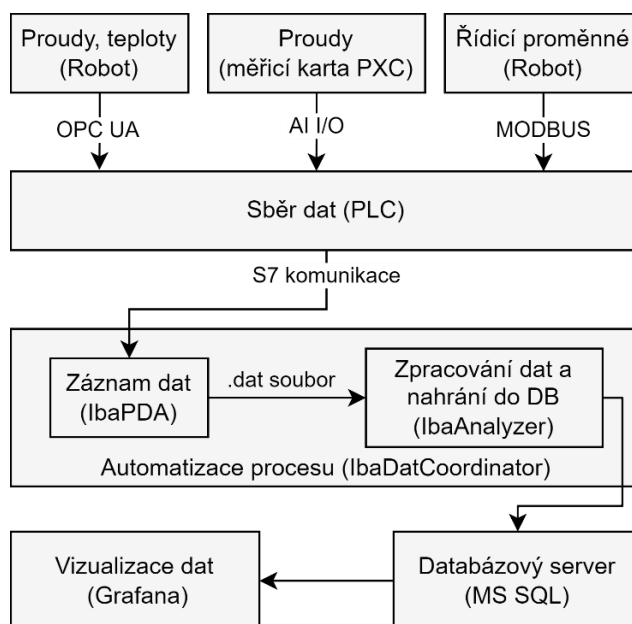
Po nahrání výsledné křivky aproximace do databáze a vykreslení přes prokládaný průběh lze lépe pozorovat trend vývoje hodnoty. V grafu jsou vykresleny aproximace za pomoci polynomů 1., 4. a 10. řádu. Z pohledu predikce stavu je nejvíce vypovídající proložení polynomem 1. řádu, tedy přímkou, která přesně udává trend vývoje průběhu. Pomocí prodloužení proložené přímky lze nalézt přesný bod, kdy při stejné tendenci vývoje dojde k protnutí kritické úrovně. Úrovně aproximace vyššího řádu více kopírují aproximovaný průběh a můžou sloužit pro zjednodušení průběhu. Při aproximaci velmi záleží na analyzovaném intervalu a hodnotách, které v daném intervalu jsou. Aproximace by měla být realizována na co největším intervalu hodnot tak, aby byl do výpočtu zanesen jak „zdravý“ stav, tak poklesy KPI indikátorů.



Obrázek 50 – Aproximace průběhu za pomoci polynomu

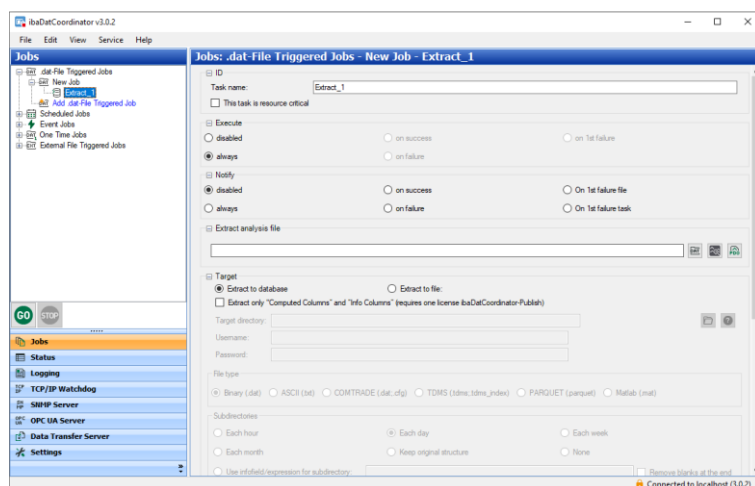
5.7 Tok dat a automatizace procesu

Proces analýzy obsahuje řadu programů, komunikací a převodů mezi formáty. Pro komplexní porozumění toku dat je proces vyobrazen v blokovém schématu (Obrázek 51). Schéma zachycuje jednotlivé kroky práce s daty a rozhraní, pomocí kterých jsou data mezi částmi zpracování předávány.



Obrázek 51 – Blokové schéma toku dat systémem

Z důvodu nutnosti realizace automatizovaného systému analýzy bylo nutné hledat řešení, které povedou k finální automatizaci celého procesu. Cílem bylo provádět automatizovaně vše od záznamu dat, přes zpracování, až k uložení do DB a vizualizaci. Samotný systém sběru dat v PLC, kde dochází k shromáždění dat z robotu, řízení a měřících karet, běží ve smyčce a není nutná jeho automatizace. Další kroky, které vedou k záznamu dat z PLC a další práce s daty již není tak snadné automatizovat. Z důvodu používání softwarů od firmy IBA lze použít pouze jedno řešení, kterým je jejich vlastní software umožňující tyto kroky automatizovat. Jmenuje se ibaDatCoordinator. Jedná se o externí aplikaci, ve které lze definovat co a v závislosti na čem se má provádět.



Obrázek 52 – Ukázka prostředí aplikace ibaDatCoordinator

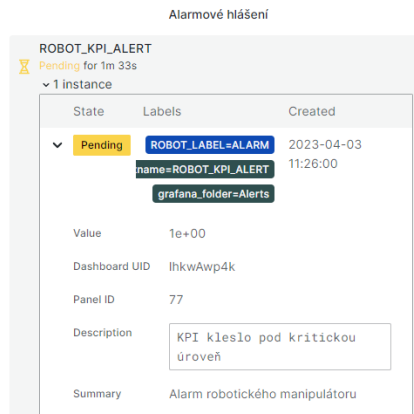
Prvním krokem pro automatizování procesu bylo nutné nastavit ibaPDA tak, aby automaticky zaznamenával data v určitých časových intervalech. Tento interval se v rámci testování měnil v závislosti na testování. Za pomoci definice tohoto intervalu je nutné záznam pustit pouze jednou a pokaždé po uplynutí onoho intervalu dojde k vytvoření .dat souboru obsahující naměřená data. Soubory se ukládají v nastaveném adresáři, z kterého jsou poté načítány pro další zpracování. IbaPDA nabízí také pokročilé způsoby triggeru, kdy lze daný interval měřit na základě podmínek signálů pro start nebo ukončení měření. Možnosti nabízejí také ukládání souborů do složek dle dnů nebo dalšího vlastního nastavení.

Dalším krokem je nutnost zpracování dat za pomoci ibaAnalyzeru. Tento proces je již nutné řešit za pomoci ibaDatCoordinator. Jelikož je cílem provést analýzu a následně data nahrát na databázový server, jedná se o úlohu Extract task. Pro tuto úlohu je nutné definovat analýzu, podle které bude zpracování dat realizováno. Jedná se o .pdo soubor, který obsahuje definici signálů výpočtů, a to včetně nastavení nahrání do DB. Druhým parametrem je nastavení cesty k adresáři, ve kterém bude automatizační software hledat nové .pdo soubory pro analýzu. Toto umístění musí být shodné jako výstup ibaPDA a software automaticky kontroluje vznik nového souboru. Po vytvoření nového souboru v adresáři dochází k okamžitému zpracování. V nastavení lze dále definovat, jak často dochází k detekci nových souborů a další upřesňující informace. Tímto krokem je vyřešena automatizace zpracování a nahrání dat do DB.

Posledním krokem je vizualizace aktuálních dat za pomoci Grafany. Tam již není nutné žádného složitějšího nastavování a stačí pouze aktivovat automatické obnovování dat. Obnovu lze nastavit na 5 s, 10 s, 15 s a tak dále.

5.8 Alarmové hlášení

Jak již bylo zmíněno v předchozí kapitole, obrazovky obsahují panel pro zobrazení nastavených alarmových hlášení. Systém alarmových hlášení je rozsáhlý a nabízí nespočet možností kategorizací alarmových hlášení, nastavení parametrů alarmů, kde se má alarm zaslat a další.



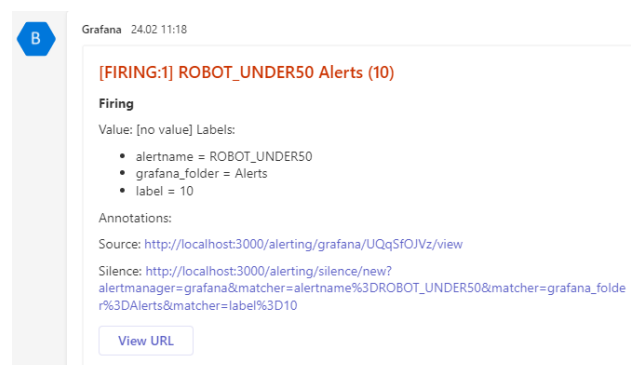
Obrázek 53 – Zobrazení panelu alarmů ve vizualizaci

Upozornění o aktivním alarmu je zobrazováno na panelech vizualizace a zobrazuje varování nebo již aktivní alarm, který je právě aktivní. Po rozbalení alarmu lze provést jeho deaktivaci nebo odložení. Pro úplné vyřešení alarmu je však nutné, aby podmínka pro jeho vyvolání pominula (např. hodnota teploty se vrátila do bezpečného intervalu).

V práci jsou definovány alarmové stavy:

- KPI robotu kleslo pod hranici 80 %
- KPI robotu kleslo pod hranici 50 %
- Teplota motoru robotu stoupla nad 90%

Upozornění o alarmu lze také přeposílat na další služby, aby došlo k informování odpovědné osoby o aktivním alarmu. Nejběžnější formou je upozornění za pomoci zaslání definovaného emailu nebo SMS zprávy. Pro funkčnost zaslání emailu je nutný funkční emailový server, který nebyl v rámci práce k dispozici. Zaslání upozornění bylo realizováno na platformu MS Teams, která nabízí jednoduché připojení ke Grafaně a zaslání upozornění za pomoci webhooku. Zásilný obsah je identický jako při zaslání mailů, pouze je cílem jiná platforma. Jak lze vidět z následujícího obrázku, ve zprávě o alarmu jsou veškeré potřebné informace o chybě a odkaz na vizualizaci.

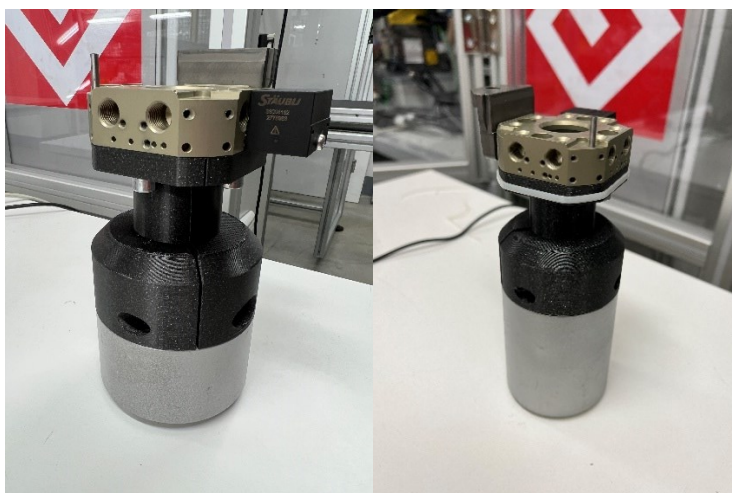


Obrázek 54 – Ukázka zaslání alarmu na službu MS Teams

6 Ověření funkčnosti systému

V průběhu práce došlo k navržení řady postupů a systémů, které mají za úkol umožnit monitoring robotického pracoviště a na základě algoritmů hodnotit jeho stav. Úkolem algoritmů je detekce možného poškození nebo degradace fyzických částí robotu. Mezi takové části patří motory, převodovky, osy a další pohyblivé části robotu. Systém analýzy vychází z předpokladu, že při zhoršení stavu robotu dojde ke zvýšení tření/vibrací nebo jinak ovlivnění mechaniky robotu, čímž dojde ke zvýšení nároků na výkon motorů. Vlivem potřeby překonat větší sílu pro stejnou operaci lze za pomoci sledování odebíraného proudu motoru změnu detekovat.

Pro ověření funkčnosti navrženého systému byl robot zatěžován závažími, která zvyšují náročnost pohybu pro motory, tedy dokáží simulovat možnou poruchu či opotřebení. Závaží jsou připevněna za pomoci vyrobeného úchytu na tool changer robotu. Tool changer slouží pro samostatnou a rychlou výměnu nástroje robotu. Pro účely testování bylo testováno zatížení robotu pomocí 2 kg a 4 kg závaží. Při zatížení nebyl v robotu nastavován žádný payload, aby nedošlo k upravení řízení motorů. Účelem měření je ověřit reakci na neočekávanou zátěž, která na osy robotu působí.



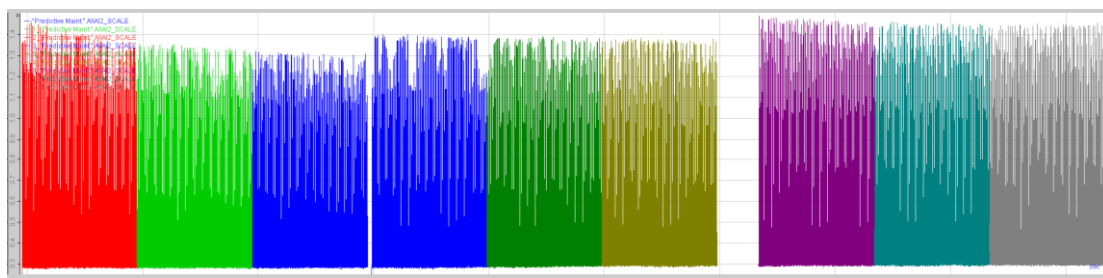
Obrázek 55 – Závaží pro simulaci zatížení robotu (2 a 4 kg)



Obrázek 56 – Zátěž připojená na efektor robotu

Pro testování byla zvolena shodná sekvence, na které byl popisován postup zpracování dat za pomoci algoritmů. Sekvence obsahuje sérii pohybů, které jsou velmi podobné pohybům, které by robot vykonával při provozu pracoviště. Pro účely testování byla snaha nastavit jednotlivé pozice robotu v sekvenci tak, aby byly zapojeny všechny osy robotu.

Interval vyhodnocování stavu robotu byl stanoven na 30 minut, během kterých robot vykonával cyklicky a pravidelně výše zmiňovanou sekvenci. Interval 30 minut stanovuje dobu, po kterou jsou data sbírána a poté zhodnocena jednou hodnotou KPI robotu, která reprezentuje stav robotu za daný interval. Proces měření probíhal za pomoci měření několika intervalů bez zatěžování, poté s 2 kg závažím a nakonec s 4 kg závažím. Celkově byly provedeny 2 měřicí „dny“, kdy první ověřil pouze reálný vliv závaží na měřený signál proudu a druhý reprezentoval delší měřicí období. Průběh proudu na jedné z os robotu v první měřicí den lze sledovat na následujícím grafu (Obrázek 57), ve kterém jsou barevně označeny zaznamenané intervaly. Měření probíhalo ve 3 fázích. V první fázi nebyl robot nijak zatěžován a byl měřen po dobu 1,5 hodiny. Měření stejné sekvence po stejně dlouhou dobu pak bylo opakováno s 2 kg a 4 kg závažím. Při pohledu na naměřený průběh lze pozorovat nárůst špiček proudu v závislosti na zvýšení zatížení robotu. Při pohledu na prvních 30 měřených minut (červený interval) lze pozorovat výskyt větších špiček než při dalším intervalu, přitom zatížení se během intervalů nijak neměnilo. Tento jev přisuzuji procesu zahřívání motorů na provozní teplotu. Při prvním intervalu došlo ke zvýšení teploty z 23 % na 33 %, která potom setrvávala až do konce měření.



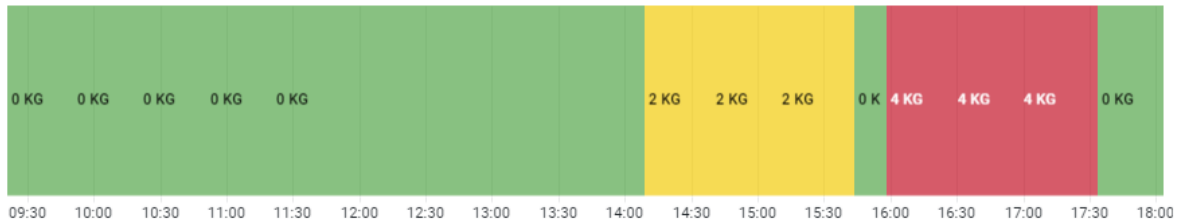
Obrázek 57 – Měření proudu při změnách zátěže robotu

V druhém měřicím „dnu“ jsem opakovал měření zcela stejným způsobem, ale celková doba měření byla prodloužena a docházelo ke měření „zdravého“ stavu také v pauzách mezi měřeními jednotlivých zátěží. Tímto byla sledována schopnost okamžitého vrácení robotu do „zdravého stavu“, pokud dojde k odebrání zátěže. Vnitřní logika kontroléru mohla při dlouhodobém zatížení upravit schéma řízení motorů, a tedy výsledek měření bez zatížení by byl odlišný. Tímto testem byla úprava řízení vyloučena. Měření probíhalo následovně:

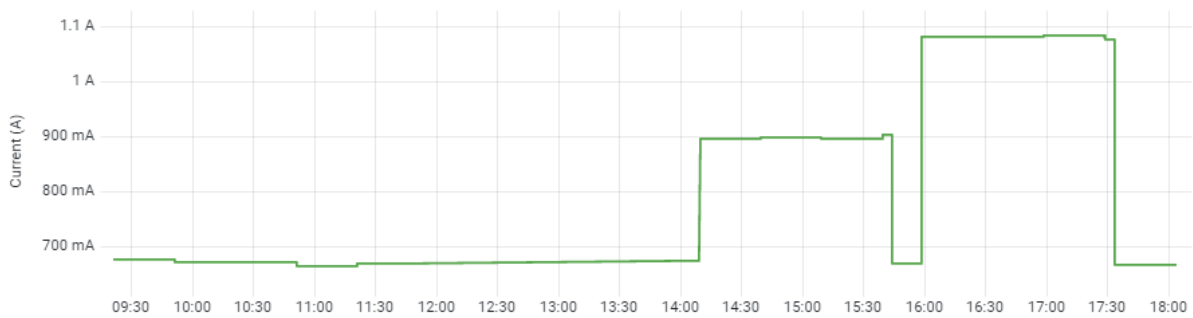
- Měření několika intervalů bez zatížení
- Měření se zatížením 2 kg
- Měření bez zatížení
- Měření se zatížením 4 kg
- Měření bez zatížení

Naměřené průběhy byly po dokončení měření vyhodnoceny a byl pozorován vliv zatížení robotu na velikost proudů os robotu. Při měření byly zaznamenávány veškeré měřené veličiny a proudy, jak získané za pomoci OPC UA komunikace, tak proudy změřené na osách robotu. V první části hodnocení vlivu zatížení na průběhy proudu dochází k analýze pouze proudů z OPC serveru, jelikož jsou dostupné pro všechny osy robotu. V další části dojde k porovnání měřených průběhů proudu na 1. ose robotu

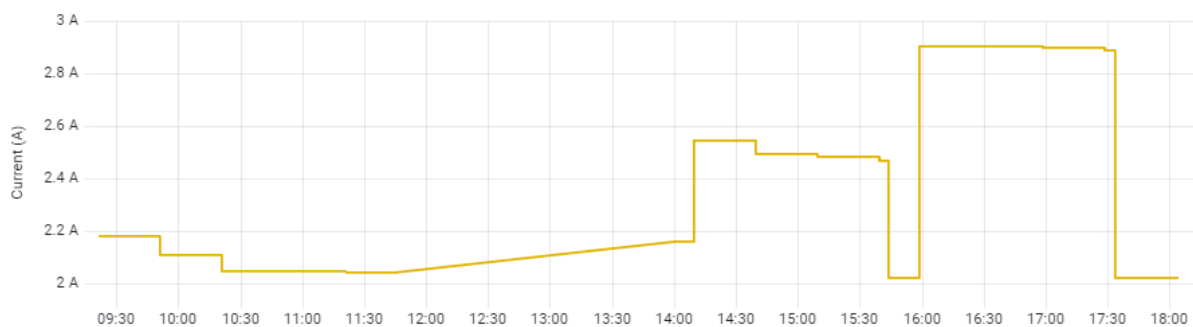
a OPC průběhů proudu 1. osy robotu. Z důvodu umístění závaží na efektoru robotu, lze předpokládat, že největší vliv bude mít zatížení na osy, které budou muset překonat gravitační a odstředivou sílu při pohybu se závažím. Jakožto vhodná osa se nabízí 2. osa robotu, která musí pohnout s prakticky celým ramenem a závaží na konci ramena způsobí posunutí těžiště dál od středu robotu. Tím dojde k znatelnému zatížení osy. Následující grafy se vztahují k 2. ose robotu. Na grafech je ukázán průběh celého měřicího dne tak, jak byl popsán výše v textu. Jednotlivé grafy reprezentují statistické hodnocení, které je počítáno ve 30 minutových intervalech. Grafy jsou exportovány z prostředí Grafana a reprezentují finální prezentaci dat ve vizualizaci.



Obrázek 58 – Stav zatížení robotu během testovacího měření



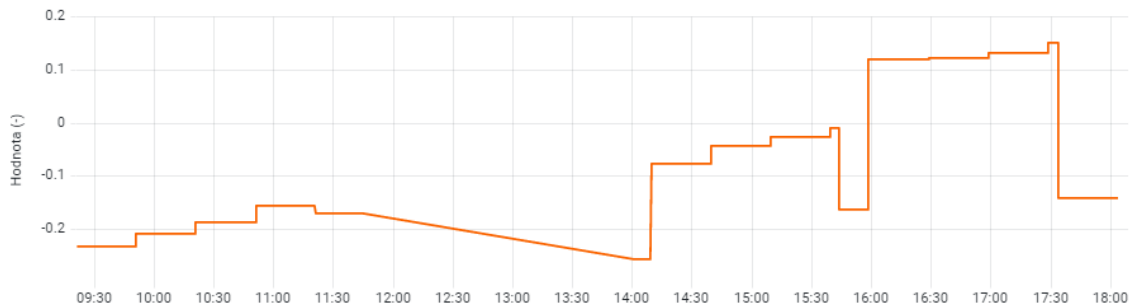
Obrázek 59 – Vliv zátěže na průměrnou hodnotu průběhu proudu 2. osy robotu



Obrázek 60 – Vliv zátěže na maximální hodnotu průběhu proudu 2. osy robotu



Obrázek 61 – Vliv zátěže na medián průběhu proudu 2. osy robotu



Obrázek 62 – Vliv zátěže na koeficient šikmosti průběhu proudu 2. osy robotu

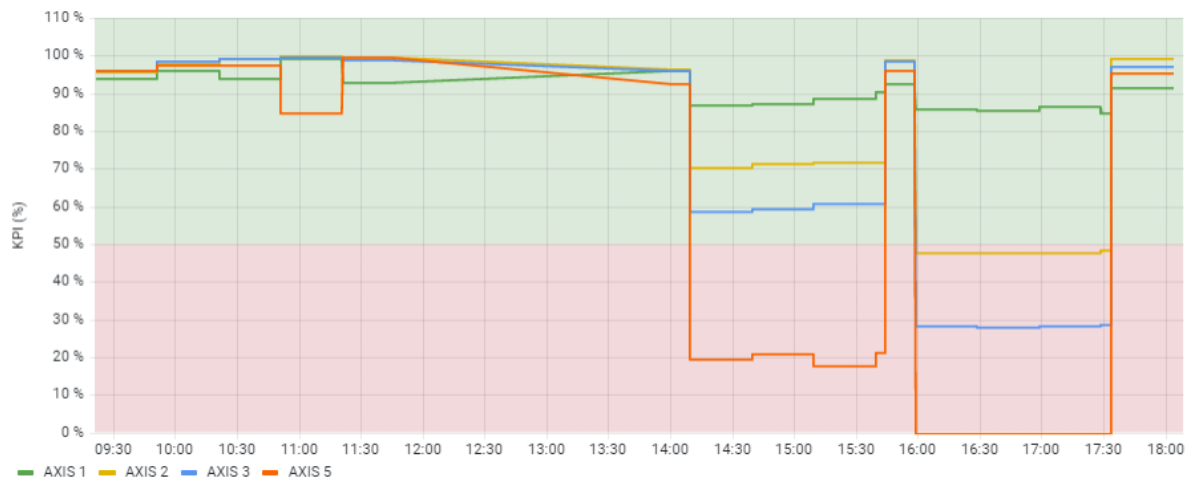


Obrázek 63 – Vliv zátěže na koeficient špičatosti průběhu proudu 2. osy robotu

Z průběhů lze velmi zřetelně pozorovat intervaly, ve kterých byl robot uměle zatěžován za pomoci přidání závaží na jeho efektor. Vliv zatížení se projevil ve všech navržených statistických hodnoceních zpracovávaného proudu robotu. Nutno zdůraznit, že těchto výsledků bylo dosaženo za pomoci OPC UA proměnných proudu komunikovaných každých 80 – 100 ms. Právě nedostatečnou rychlost komunikace považuji za důvod nepřesností způsobené při opakovaném měření stejné sekvence robotu. Největší kolísání hodnot stejného průběhu lze pozorovat u maximální hodnoty proudu (Obrázek 60) nebo u koeficientu špičatosti (Obrázek 63). Právě maximální hodnota odpovídá proudovým špičkám zaznamenaného proudu, a při nedostatečném vzorkování měřeného proudu dochází k unikům informací o špičkách. I přes relativně pomalé vzorkování je stále údaj o maximální hodnotě průběhu jeden z nejvíce reprezentujících zatížení motorů. Společně s průměrnou hodnotou jsou řazeny jako hlavní dvojice statistických hodnocení, ze kterých lze pozorovat změnu zatížení robotu a při opakovaných pokusech jsou stabilní. Následující KPI hodnocení vycházejí pouze ze statistického hodnocení maximální a průměrné hodnoty. Ostatní hodnocení byly u některých os nedostatečným vzorkováním natolik kolísavé, že velmi výrazně ovlivňovaly výsledné KPI.

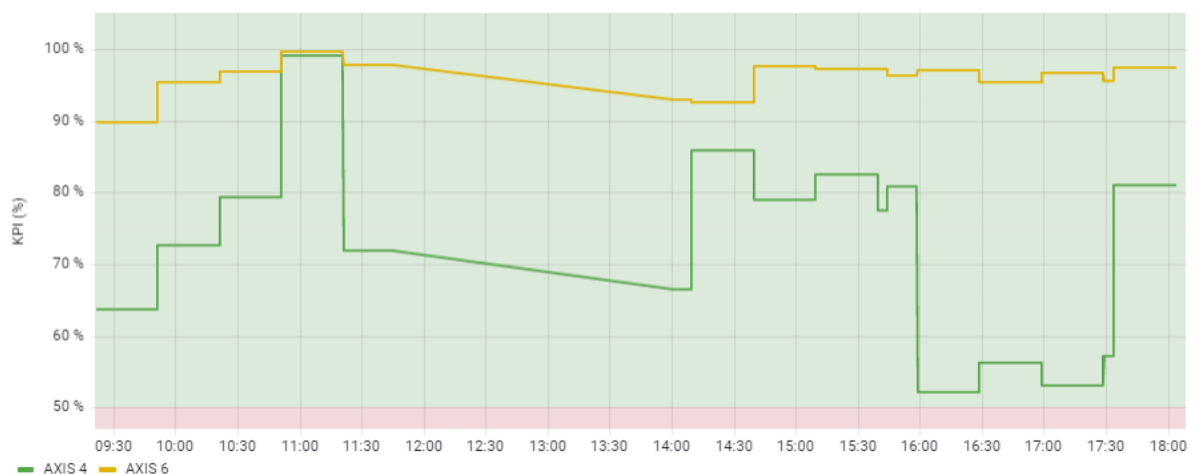
Po provedené analýze dat a výpočtu KPI hodnot statistických hodnocení došlo k výpočtu KPI indikátoru celé osy robotu. Sjednocení KPI indikátorů bylo provedeno za pomoci metody průměrování.

Výsledky výpočtu KPI indikátoru jednotlivých os robotu lze sledovat na následujícím grafu (Obrázek 64). V grafu jsou zobrazeny vybrané osy robotu, u kterých dochází k předpokládanému poklesu KPI při zatížení robotu a vykazují stabilní hodnoty KPI během opakovaného měření průběhu za stejných podmínek. Jedná se o osy robotu 1, 2, 3 a 5, které musí vyvinout větší sílu na pohyb ramenem. Největší vliv mělo zatížení na osy 3 a 5, které vyvíjejí zdvih ramene robotu společně se zátěží.



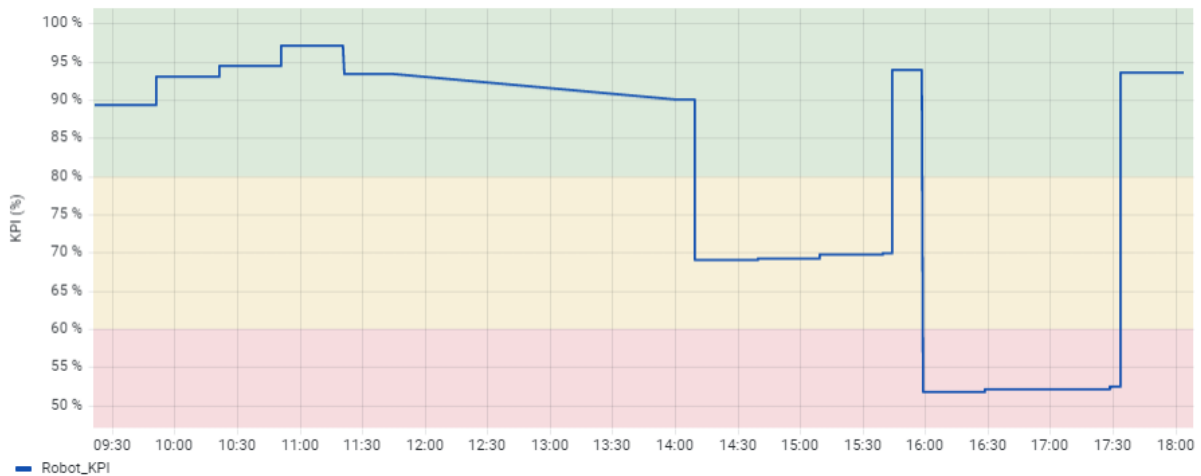
Obrázek 64 – KPI hodnocení průběhu proudu robotu na vybraných osách

Při pohledu na KPI hodnocení zbylých 2 os nelze přímo pozorovat znatelné intervaly, kdy došlo k zatížení robotu a také jejich hodnota KPI velmi kolísá. Menší reakci na zatížení robotu si lze logicky vysvětlit. Osa 4 rotuje část ramene a nevyvíjí sílu ve velké závislosti na aplikovaném zatížení, průběh měřených proudů se tedy tolik neovlivní. 6. osa robotu nezaznamenala prakticky žádný vliv zatížení a opět je při pohledu na osy robotu patrné, že slouží pouze pro otáčení efektoru kolem své osy, tedy opět nevykonává žádné fyzikálně namáhavé pohyby. Kolísání KPI hodnot při opakování stejného průběhu si vysvětlují nedostatečnou vzorkovací frekvencí měřeného proudu.



Obrázek 65 – KPI hodnocení průběhu proudu robotu na 4. a 6. ose

Z KPI hodnocení jednotlivých os dochází poté k výpočtu finálního KPI hodnotící stav celého robotu. To je složeno z KPI všech os robotu a reprezentuje stav robotu jako celku. Průběh lze sledovat na následujícím obrázku (Obrázek 66).



Obrázek 66 – Celkové KPI hodnocení robotu

Vizualizované KPI hodnocení robotu vychází z KPI hodnocení jednotlivých os. KPI os byly sjednocovány za pomoci jejich průměrování. Pokud by došlo k poškození jedné osy, na finální výsledek KPI by neměl tak znatelný vliv, jako při jejich násobení. Sjednocení KPI za pomoci násobení bylo realizováno, ale po sériích testování byla nutná jeho úprava. Sjednocení násobením probíhalo jak při sjednocení KPI hodnocení jedné osy (tedy násobení 5 KPI), tak při sjednocení hodnocení všech os do jednoho KPI (vzájemné násobení 6 KPI). Pro příklad si lze představit, že každá osa je hodnocena KPI hodnotou 90 %. Při sjednocení do jednoho KPI vychází:

$$0,9^6 \cdot 100 = 53,14 \%$$

Pokud výsledné KPI všech os vyjde 90 %, které lze stále považovat za velmi zdravý stav, výsledné KPI robotu bude pouze 53 %. Pokud by bylo KPI os blíže zdravé hodnotě 100 %, pokles nebude takto výrazný. Vlivem kolísání měřené hodnoty proudu nelze zajistit KPI hodnotu blízkou 100 %. Opět zde sledávám velký vliv nedostatečné vzorkovací frekvence, která způsobuje kolísání výsledných hodnocení průběhu. Z tohoto důvodu bylo sjednocení za pomoci násobení KPI hodnot mezi sebou upraveno na sjednocení za pomoci průměrování.

Výše uvedené průběhy a KPI hodnocení os, potažmo robotu, vždy vycházely z hodnot proudů získaných z robotu za pomoci OPC UA komunikace. Proud na 1. ose robotu byl ale měřen také za pomoci měřicí karty přímo na přívodním kabelu robotu. Karta měřila v intervalech 200 ms vlivem reakční doby na skokovou změnu měřeného průběhu. Měřený průběh byl zpracován za pomoci shodných algoritmů a následující grafy zobrazují porovnání měřených průběhů a výsledných výpočtů KPI hodnot.

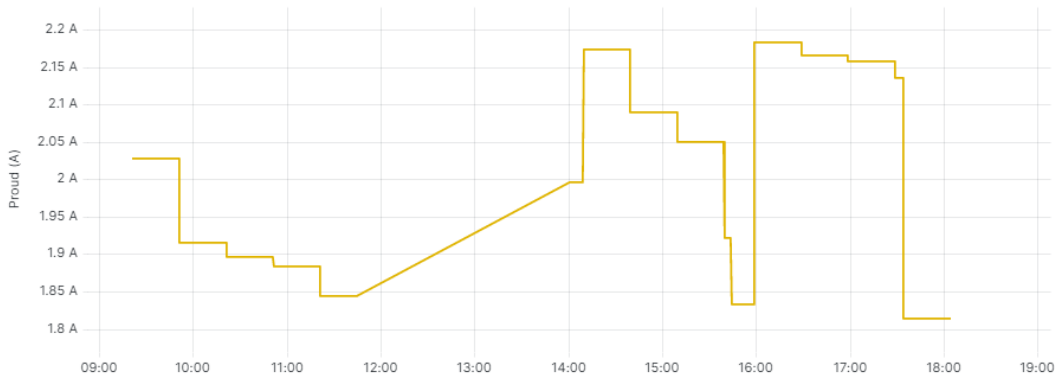
Z měřených průběhů lze sledovat, že vliv zatížení na 1. osu robotu není ze statistických hodnocení průběhu za pomoci průměrné hodnoty příliš zřetelný. V porovnání OPC UA hodnota oproti měřené hodnotě lze z mého pohledu měřený průběh považovat za horší nežli přes OPC UA. Při porovnání průběhů maximálních proudů na ose, lze pozorovat změnu pouze ve velikosti proudu, ale reakce na zvýšení zátěže robotu je shodná.



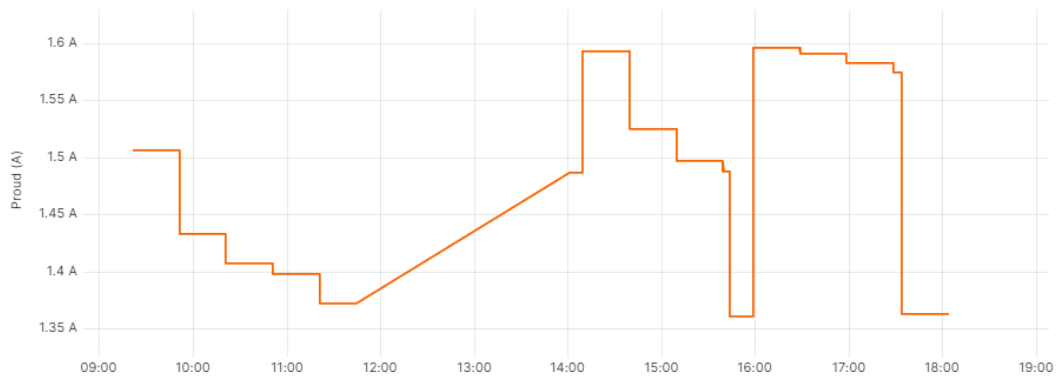
Obrázek 67 – Vliv zátěže na průměrnou hodnotu OPC průběhu proudu 1. osy robotu



Obrázek 68 – Vliv zátěže na průměrnou hodnotu měřeného průběhu proudu 1. osy robotu

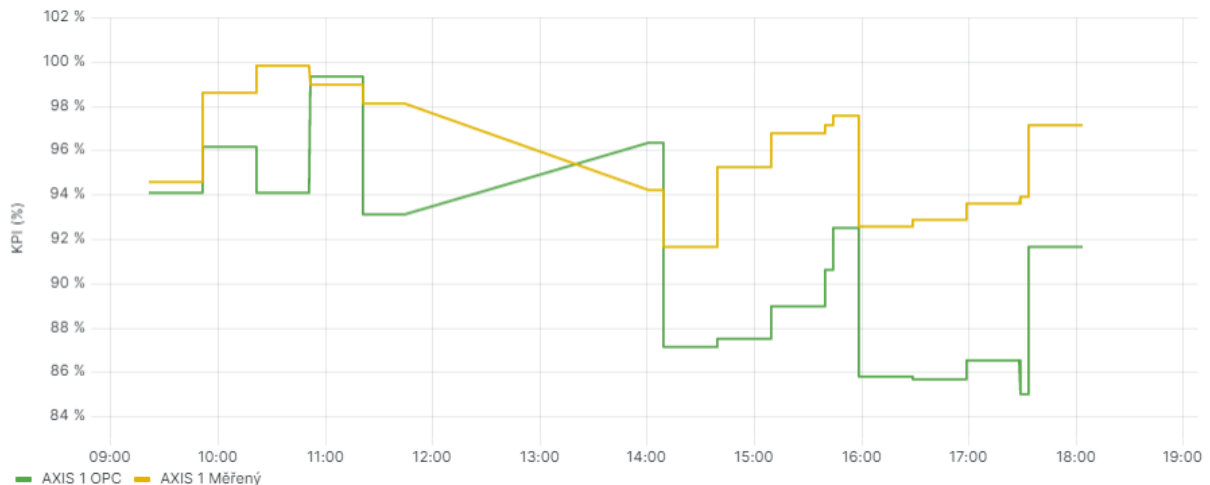


Obrázek 69 – Vliv zátěže na maximální hodnotu OPC průběhu proudu 1. osy robotu



Obrázek 70 – Vliv zátěže na maximální hodnotu měřeného průběhu proudu 1. osy robotu

Při výpočtu KPI hodnot z měřených veličin proudu lze pozorovat menší indikaci změny u proudu měřeného za pomoci vlastní karty, oproti analýze proudu získaného přes OPC UA komunikace. Tento jev bych připsal také skutečnosti, že měřicí karta měří absolutní hodnotu proudu na motoru, oproti hodnotě získané z robotu, která obsahuje informaci o znaménku, neboli směru pohybu robotu. Proudů u ostatních os nebyly z důvodů omezení vybavení pracoviště měřeny, takže je nelze porovnat. Z výsledků porovnání lze přesto předpokládat, že rozdíly budou obdobné a zmenšením vzorkovací frekvence díky omezené schopnosti karty, dojde k zhoršení stability průběhu a reakce na změnu stavu robotu.



Obrázek 71 – Porovnání analýzy stavu osy robotu za pomoci měřeného proudu a proudu z OPC

Závěr

Cílem práce je monitorování provozu a návrh systému prediktivní údržby pro robotické pracoviště. Práce byla realizována na pracovišti TQC testeru, který obsahuje dvojici robotů Fanuc. Funkcí pracoviště je provádění automatizované kamerové kontroly snímaného objektu. Systém sledování robotu byl navržen a vyvíjen pro robot Fanuc LR Mate 200iD. Součástí navrženého systému jsou jednotlivé kroky práce s daty od sběru, až po archivaci a vizualizaci.

Systém monitorování se zaměřuje na poruchy vzniklé mechanickým poškozením nebo jiným ovlivněním zatížení robotu. Pro návrh systému monitorování stavu robotu bylo vycházeno z předpokladu, že při rostoucím opotřebením nebo poškození částí robotu dojde ke vzniku většího odporu (tření) mechanických částí a motory v osách robotu budou muset pro překonání většího odporu vyvinout větší výkon. Analyzované veličiny vycházejí z metody prediktivní údržby, která se zabývá monitorováním výkonnosti a provozních hodnot stroje. Z důvodu sledování mechanického poškození robotu a změnách chování jeho motorů, jako hlavní analyzovanou veličinou byl zvolen proud. Proud se mění v závislosti na výkonu motoru a umožňuje monitorování jeho zatížení. Proud byl zvolen také z důvodu, že se jedná o nejlépe dostupnou veličinu reprezentující zátěž robotu, kterou lze z robotu vyčítat za pomoci komunikačního rozhraní.

Sběr dat je realizován řídicím PLC robotického pracoviště, za pomoci kterého dochází ke sjednocení všech sledovaných veličin robotu. Hodnoty proudu z jednotlivých os jsou čteny za pomoci OPC UA komunikace v intervalech 80 – 100 ms. Součástí komunikace je také přenos aktuální rychlosti robotu. Další provozní data robotu je nutné číst přes Modbus komunikaci, která slouží pro řízení robotu pomocí PLC. Přes Modbus komunikaci jsou zasílány informace, zda je robot spuštěn, zda vykonává program, jaké je číslo vykonávaného programu a další řídicí proměnné. Ty jsou při analýze potřeba z důvodu selekce analyzovaného intervalu. Proměnné přes Modbus jsou posílány co 1 – 3 sekundy. Proměnné jsou posílány přes Modbus z důvodu, že přímo v OPC UA protokolu robotu nelze definovat vlastní proměnné pro přenos a nabízí pouze základní předem nastavené proměnné. Přes OPC lze přenášet uživatelsky definované proměnné, ale pouze za pomoci definice v Modbus data modelu, který se následně posílá přes OPC UA. Problém Modbus data modelu spočívá v nutnosti komunikace velké datové oblasti, která způsobuje výrazné výpadky a zpomalení komunikace. Z tohoto důvodu nebyla komunikace Modbus data modelu přes OPC UA použita. Kvůli snaze o nezávislost systému na výrobcu, typu a dostupných proměnných robotu, bylo realizováno měření proudů os na přívodním kabelu robotu. Měření bylo provedeno na 1. ose a provedeno za pomoci měřicího proudového transformátoru Phoenix Contact. Trafo bylo připojeno k PLC na vysokorychlostní (HS) analogovou kartu. V testovací fázi měření bylo vybíráno z dvojice výrobců měřících transformátorů a dvojice modelů analogové PLC karty. Na základě testování byla vybrána dříve zmíněná kombinace (PXC + HS AI).

Pro záznam dat z PLC byl zvolen nástroj ibaPDA. Jedná se o software pro PC, který umožňuje záznam dat přímo z datových bloků Siemens PLC za pomoci S7 komunikace. Zaznamenávány jsou veškeré dříve zmíněné proměnné z robotu a měřicí karty. Data jsou zaznamenávána po dobu definovaného intervalu a ukládána v datovém binárním souboru .dat.

Soubor .dat je následně zpracováván za pomoci softwaru ibaAnalyzer, který je použit pro provedení analýzy dat. Jeden načtený datový soubor představuje jedno analyzované období, které bude ve výsledku popsáno jednou hodnotou stavu robotu. V souboru záznamu je selektován pouze interval, kdy robot vykonává žádanou sekvenci a dochází k jeho hodnocení za pomoci statistických

veličin průměrné hodnoty, maxima, mediánu, koeficient šikmosti a špičatosti. Po výpočtu statistických hodnocení analyzovaného období dochází k porovnání aktuální hodnoty se „zdravou“ hodnotou, která byla naměřena na začátku provozu robotu. Výsledkem porovnání je KPI indikátor, který reprezentuje podobnost se „zdravou“ hodnotou v rozmezí 0 – 100 %, kde 100 % reprezentuje největší shodu. KPI jsou počítány pro jednotlivé statistické hodnocení zvlášť a následně za pomoci průměrování sjednoceny do jednoho finálního KPI indikátoru osy. Stejným způsobem jsou poté sjednoceny KPI os do hodnocení celkového stavu robotu. Sjednocení bylo testováno také za pomoci vzájemného násobení KPI mezi sebou, ale z důvodu kombinace mnoha hodnot nebylo použito. Po dokončení analýzy dochází k extrakci výsledků na databázový server MS SQLEXPRESS, kde jsou data dlouhodobě ukládána.

Databáze slouží jako uložení dat pro jejich následné čtení do vizualizace, která zobrazuje výsledky analýz a historický pohled na veškeré zaznamenané proměnné robotu. Vizualizace je realizována za pomoci nástroje Grafana a obsahuje dvojici obrazovek. První obrazovka zobrazuje základní informace o stavu obou robotů a slouží pro rychlý přehled o jejich stavu. Druhá obrazovka již zobrazuje konkrétní informace o zvoleném robotu a obsahuje veškerá data, která jsou o robotu dostupná. Nabízí pokročilé informace o aktuálním a historickém stavu robotu nebo konkrétní informace o zvolené ose. V historických grafech lze pozorovat vývoj původně naměřených dat, veškeré provedené mezivýpočty a výsledné průběhy KPI indikátorů. Součástí vizualizace je systém alarmů, který detekuje pokles KPI robotu pod definovanou úroveň a provede nastavené upozornění. Vizualizace obsahuje také systém predikce stavu, který funguje na principu proložení průběhu KPI za pomoci polynomu x -tého řádu a nabízí lepší vizualizaci trendu KPI hodnoty.

K ověření funkčnosti systému došlo za pomoci přidání neočekávané zátěže na robota, která simulovala možné zhoršení stavu nebo poškození robotu. Závaží bylo umístěno na efektor a pro motory robotu představovalo přidanou zátěž, která by nastala při mechanickém poškození částí robotu. Vlivem zatížení došlo k ovlivnění proudových odběrů motorů. Měření vlivu zatížení na změnu KPI hodnot (stavu) robotu probíhalo ve 3 fázích, které reprezentují postupné zatěžování robotu. Během všech měření robot vykonával cyklicky jeden program. Měření bylo prováděno po 30 minutových intervalech a opakováno vždy minimálně 3x pro každou fázi. Nejprve byl sledován stav robotu bez zatížení a pozorováno, zda bude při opakujícím se měření stav robotu hodnocen pořád stejně. Výsledné KPI robotu bez zatížení se pohybovalo v rozmezí 88 – 98 %. Znatelné vychýlení KPI nastalo u prvního měřeného intervalu, které přisuzují procesu zahřívání motorů na provozní teplotu. Po opakování série měření stavu robotu bez zátěže bylo na efektor robotu umístěno 2 kg závaží, s kterým byla série měření opakována. Na základě následné analýzy dat došlo k ovlivnění průběhů všech statistických hodnocení. Stejný proces se opakoval s větším 4 kg závažím, které opět výrazně ovlivnilo statistické hodnocení a výsledné KPI indikátory. Mezi simulovanými zatíženími byl robot provozován opět bez závaží, aby došlo k ověření, že lze ihned indikovat „zdravý“ stav, a robot neupravil své řízení pro práci se zátěží. Během testů nebyl v nastavení robotu definovaný žádný payload, aby nedošlo ke změně řízení robotu. Na základě testů nebyla žádná změna řízení detekována a KPI hodnota se vždy vrátila do „zdravého“ stavu ihned po odebrání zátěže. Zatížení robotu mělo největší vliv na 1., 2., 3. a 5. osu, u kterých došlo k výraznému poklesu KPI indikátoru reprezentující jejich stav. Z výsledného průběhu KPI robotu lze na první pohled sledovat okamžiky, kdy došlo k zatížení. Bez zatížení se KPI robotu pohybovalo nad 90 %, při zatížení 2 kg závažím kleslo na 70 % a při zatížení 4 kg závažím kleslo až na 55 %. Obdobné výkyvy lze pozorovat také na KPI indikátorech jednotlivých os robotu. Na základě tohoto testu lze považovat

navržené řešení za funkční, schopné rozpoznat změnu průběhů proudů na motorech robotu a na základě porovnání se „zdravou“ hodnotou určit celkový stav robotu. Výsledkem analýzy je zobrazení aktuálního stavu, historie stavu a dalších provozních hodnot robotu ve vizualizaci. Těchto výsledků bylo dosaženo za pomoci proudů získaných z robotu pomocí OPC UA komunikace.

V další části práce došlo k porovnání výsledků analýz, které byly dosaženy na základě analýzy proudů z OPC UA komunikace a proudů měřených za pomoci proudového transformátoru. Výsledné KPI robotu zpracované stejnými postupy bylo více vypovídající o zatížení robotu při použití dat z OPC UA komunikace. Při porovnání maximálních hodnot proudů dojde k téměř shodné indikaci změny stavu. Při porovnání dalších statistických hodnocení jsou výsledky z měřicí karty méně vypovídající o změnách stavu. Tento výsledek lze vysvětlit absencí polarity proudu a nižší vzorkovací frekvencí měřicí karty PXC.

Oblast pro další vývoj navrženého systému vidím především ve zrychlení komunikace mezi PLC a robotem, která je hlavním nedostatkem. Jelikož při zpracování proudů dochází k zpracování maximálních hodnot a statistických hodnocení závislých na rozložení hodnot v intervalu, je nutné zdrojový proud zaznamenávat za pomoci co nejvyšší vzorkovací frekvence (jednotky ms). V opačném případě dochází ke ztrátě informací o proudových špičkách, nebo jiných krátkodobých změnách průběhů proudů. Obdobným vylepšením by bylo vybavení pracoviště měřicí kartou, která je schopná záznamu proudu s vzorkováním v rámci milisekund a dokáže zachytit proudové špičky při pohybu robotu. Další omezení systému vidím v přístupnosti analyzovaných dat z robotu. Robot poskytuje základní proměnné a veličiny o jeho provozu. Pro realizaci komplexního systému sledování stavu robotu by byl nutný přístup k interním a zdrojovým datům z měřicích prvků robotu. Z pohledu sledování robotu Fanuc jsou takové proměnné často dostupné pouze pro interní systémy v řídicí jednotce robotu a nelze je komunikovat „ven“. Pro další vývoj by bylo nutné vybavení robotu vlastními senzory pro měření provozních veličin, nebo nalezení řešení pro čtení interních systémových proměnných z řídicí jednotky robotu.

Literatura

- [1] ČSN EN 13306. *Údržba - Terminologie údržby*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2018.
- [2] LEGÁT, Václav. *Management a inženýrství údržby*. 1. vyd. [Praha]: Professional Publishing, 2013. ISBN 978-80-7431-119-2.
- [3] PRADHAN, M.K. *Trends and Perspectives in Industrial Maintenance Management* [online]. 2006 [cit. 2022-10-31]. Dostupné z: <http://dspace.nitrkl.ac.in/dspace/handle/2080/786>
- [4] MOUBRAY, John. *Reliability-centred maintenance*. 2nd ed. Oxford: Butterworth-Heinemann, 2007. ISBN 0750633581.
- [5] *Systém údržby*. 1. vydanie. Bratislava: CKV Consult, s r.o., 2015. ISBN 978-80-971986-0-2.
- [6] MERKT, Oana. *On the Use of Predictive Models for Improving the Quality of Industrial Maintenance: an Analytical Literature Review of Maintenance Strategies* [online]. 01-04 September 2019. Leipzig, Germany: IEEE, 2019 [cit. 2023-04-24]. ISBN 978-83-952357-8-8. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8860031>
- [7] KRUPA, Miroslav. Technická prognostika v kontextu prediktivní údržby. *AUTOMA* [online]. 2012, 2012(2), 4 [cit. 2023-04-24]. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/9344.pdf
- [8] HASHEMIAN, H. a Wendell BEAN. State-of-the-Art Predictive Maintenance Techniques*. *IEEE Transactions on Instrumentation and Measurement* [online]. 2011, 60(10), 3480-3492 [cit. 2022-10-31]. ISSN 0018-9456. Dostupné z: doi:10.1109/TIM.2009.2036347
- [9] SCHEFFER, Cornelius. *Practical Machinery Vibration Analysis and Predictive Maintenance* [online]. 1st. Newnes, 2004 [cit. 2023-04-24]. ISBN 9780080480220. Dostupné z: <https://www.elsevier.com/books/practical-machinery-vibration-analysis-and-predictive-maintenance/scheffer/978-0-7506-6275-8>
- [10] MERIZALDE, Yuri, Luis HERNÁNDEZ-CALLEJO a Oscar DUQUE-PEREZ. State of the Art and Trends in the Monitoring, Detection and Diagnosis of Failures in Electric Induction Motors. *Energies* [online]. 2017, 10(7) [cit. 2022-10-31]. ISSN 1996-1073. Dostupné z: doi:10.3390/en10071056
- [11] LEANDRO, Erik, Levy LACERDA DE OLIVEIRA, Jonas DA SILVA, Germano LAMBERT-TORRESM a Luiz DA SILV. Predictive Maintenance by Electrical Signature Analysis to Induction Motors. In: ARAJO, Rui Esteves, ed., Rui ARAJO. *Induction Motors - Modelling and Control* [online]. InTech, 2012 [cit. 2022-10-31]. ISBN 978-953-51-0843-6. Dostupné z: doi:10.5772/48045
- [12] LU, Bin, David DUROCHER a Peter STEMPEL. Predictive maintenance techniques. *IEEE Industry Applications Magazine* [online]. 2009, 15(6), 52-60 [cit. 2022-10-31]. ISSN 1077-2618. Dostupné z: doi:10.1109/MIAS.2009.934444
- [13] WU, Hong a Jiasi FENG. A Review of Fault Diagnosis Methods of Robot Joint Servo System. In: *2020 IEEE International Conference on Mechatronics and Automation (ICMA)* [online]. IEEE, 2020, s. 61-66 [cit. 2023-03-06]. ISBN 978-1-7281-6416-8. Dostupné z: doi:10.1109/ICMA49215.2020.9233583

- [14] Servo Motor and Drive Troubleshooting Guide. In: *Global electronics services inc.* [online]. [cit. 2023-03-06]. Dostupné z: <https://gesrepair.com/servo-motor-drive-troubleshooting-guide/>
- [15] KUKA iiQoT: Přidané hodnoty založené na datech prostřednictvím IIoT pro vaše roboty. In: *KUKA* [online]. [cit. 2023-04-27]. Dostupné z: <https://www.kuka.com/cs-cz/produkty,-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/software/software-cloud/iiqot>
- [16] iba hardware. In: *Iba ag* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.iba-ag.com/en/hardware>
- [17] Predictive Maintenance through condition monitoring. In: *OMRON* [online]. [cit. 2023-03-06]. Dostupné z: <https://omron.showpad.com/share/OoERWwM9aNv95JTngzclj/0>
- [18] ibaPDA: The scalable Core Software in Measuring Value Acquisition. In: *Iba AG* [online]. [cit. 2023-04-02]. Dostupné z: <https://www.iba-ag.com/en/ibapda>
- [19] S7comm. In: *Wireshark* [online]. [cit. 2023-04-02]. Dostupné z: <https://wiki.wireshark.org/S7comm>
- [20] VOJÁČEK, Antonín. Průmyslová komunikace OPC UA - 1.díl - popis protokolu. In: *Automatizace.hw* [online]. [cit. 2023-04-02]. Dostupné z: <https://automatizace.hw.cz/prumyslova-komunikace-opc-ua-1dil-popis-protokolu.html>
- [21] Product Details - Industry Mall: 6ES7134-6HB00-0DA1. In: *Siemens* [online]. [cit. 2023-03-06]. Dostupné z: <https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/6ES7134-6HB00-0DA1>
- [22] Product Details - Industry Mall: 6ES7134-6HD01-0BA1. In: *Siemens* [online]. [cit. 2023-03-06]. Dostupné z: <https://mall.industry.siemens.com/mall/cs/cz/Catalog/Product/6ES7134-6HD01-0BA1>
- [23] Fanuc LR Mate 200iD. In: *Fanuc* [online]. [cit. 2023-04-02]. Dostupné z: <https://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/%C5%99ada-lrmate/lrmate-200-id>
- [24] Current transducers - MCR-S-1-5-UI-DCI. In: *Phoenix Contact* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.phoenixcontact.com/en-pc/products/current-transducers-mcr-s-1-5-ui-dci-2814634>
- [25] Monitoring relays E83-2050. In: *Carlo Gavazzi* [online]. [cit. 2023-03-06]. Dostupné z: <https://www.gavazziautomation.com/nsc/HQ/EN/product/E83-2050>
- [26] KONDRATH, N. a M.K. KAZIMIERCZUK. Bandwidth of Current Transformers. *IEEE Transactions on Instrumentation and Measurement* [online]. 2009, 58(6), 2008-2016 [cit. 2023-03-06]. ISSN 0018-9456. Dostupné z: doi:10.1109/TIM.2008.2006134
- [27] HENDL, Jan. *Přehled statistických metod zpracování dat: analýza a metaanalýza dat*. Vyd. 2., opr. Praha: Portál, 2006. ISBN 80-7367-123-9.
- [28] IZAGIRRE, Unai, Imanol ANDONEGUI, Itziar LANDA-TORRES a Urko ZURUTUZA. A practical and synchronized data acquisition network architecture for industrial robot predictive maintenance in manufacturing assembly lines. *Robotics and Computer-Integrated Manufacturing* [online]. 2022, 74 [cit. 2023-03-30]. ISSN 07365845. Dostupné z: doi:10.1016/j.rcim.2021.102287

Seznam příloh

- Příloha I** **Příloha v IS EDISON**
Obsah: Projekt v TIA Portalu obsahující PLC program pro sběr dat z pracoviště.
- Příloha II** **Příloha v IS EDISON**
Obsah: Ukázková data robotu zaznamenaná během jedné měřicí sekvence robotu.
- Příloha III** **Příloha v IS EDISON**
Obsah: Ukázková data robotu zaznamenaná při dlouhodobějším měření.
- Příloha IV** **Příloha v IS EDISON**
Obsah: Soubor analýzy dat pro výpočet KPI indikátorů robotu.
- Příloha V** **Příloha v IS EDISON**
Obsah: Soubor analýzy dat pro aproximaci průběhu KPI indikátorů robotu.
- Příloha VI** **Příloha v IS EDISON**
Obsah: SQL dotaz pro načtení průběhu KPI indikátoru robotu z databáze.
- Příloha VII** **Příloha v IS EDISON**
Obsah: Export obrazovky vizualizace zobrazující základní informace o více robotech.
- Příloha VIII** **Příloha v IS EDISON**
Obsah: Export obrazovky vizualizace zobrazující podrobné informace o robotu.