

# MPO TRIO

## Název projektu:

Nová technologie pro inteligentní plánování pohybu robotů v průmyslových procesech  
(FV 20597)

Software

## Software interfacu s průmyslovým robotem (Implementační a uživatelská dokumentace)

Jan Reitinger (ZČU), Vlastimil Šetka (ZČU), Martin Švejda (ZČU)

18. března 2020



MINISTERSTVO  
PRŮMYSLU A OBCHODU

## Identifikátor účastníka projektu:

ZČU: Západočeská univerzita v Plzni, LT: LaserTherm spol. s r.o.

## Popis úkolu z přihlášky projektu

Porozumění v oblasti transferu dat plánovaných trajektorií do řídicích systémů průmyslových robotů. Možnost implementovat interface pro světové výrobce průmyslových robotů vedoucí na možnost nasazení vyvinutého záznamového zařízení na libovolné typy průmyslových robotů.

## Úvod

Předložená dokumentace se zabývá dosaženými výsledky při návrhu a realizaci softwaru interfacu mezi vyvíjeným trekovacím zařízením a kontrolerem průmyslového robotu. Dokumentace dále rozšiřuje popis činností zahájených v roce 2019 a dokumentovaných v technické zprávě [1], zabývající se především propojením s průmyslovými roboty UR a Stäubli.

Vzhledem na zacílení průmyslového partnera LT na roboty od firmy Fanuc, které dlouhodobě využívají ve svých aplikacích, je předložená dokumentace věnována výhradně implementačním poznámkám pro komunikaci s kontrolery robotů Fanuc z nadřazeného SW řízení pohybu.

Vzhledem k podstatě SW, který je dílčí prerekvizitou hlavních výstupů projektu *[Gprot] Prototyp záznamového zařízení a [Gprot] Prototyp robotické buňky osazené záznamovým zařízením (pilotní aplikace)*, je podrobně zahrnuta implementační část a část testování (viz samostatný dokument [2]). Část uživatelské dokumentace bude podrobně popsána současně s celkovým SW vybavením a uživatelským prostředím kompletního trekovacího zařízení.

## Možnosti propojení s roboty Fanuc

Přesto, že roboty Fanuc patří k neznámějším a nejvíce používaným robotům napříč všemi odvětvími, jejich řídicí systém (kontroler) je poměrně omezený. Pohybové funkce robotu jsou dostupné výhradně přes integrované programovací prostředí TP, které je možné doplnit řadou výrobcem poskytovaných rozšiřujících balíčků (Options).

Navíc výrobce u některých Options vyžaduje speciální kvalifikaci zákazníka a konkrétního projektu, případně je na některých trzích vůbec nedodává.

Na základě průzkumu možností povelování robotů a konzultací s odborníky z řad integrátorů byly specifikovány 2 možné přístupy:

- Offline povelování robotu založené na option SPLINE MOTION - podrobně popsáno dále
- Online korekce pohybové trajektorie založené na option DYNAMIC PATH MODIFIER - možnosti diskutovány dále

Kromě uvedených dvou možností byla prověřována ještě funkce STREAM MOTION, která umožňuje přímé řízení os robotu v polohovém režimu z vnějšího systému s minimálními omezeními. Nicméně toto je právě příklad funkce, kterou výrobce na evropském trhu neposkytuje.

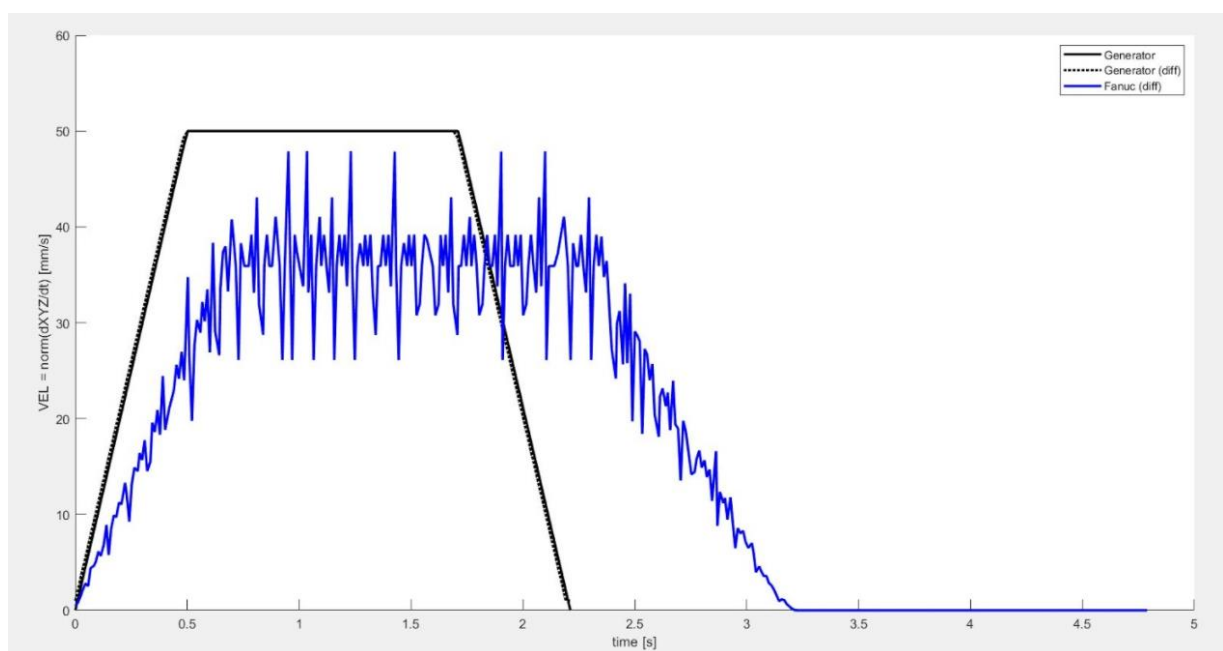
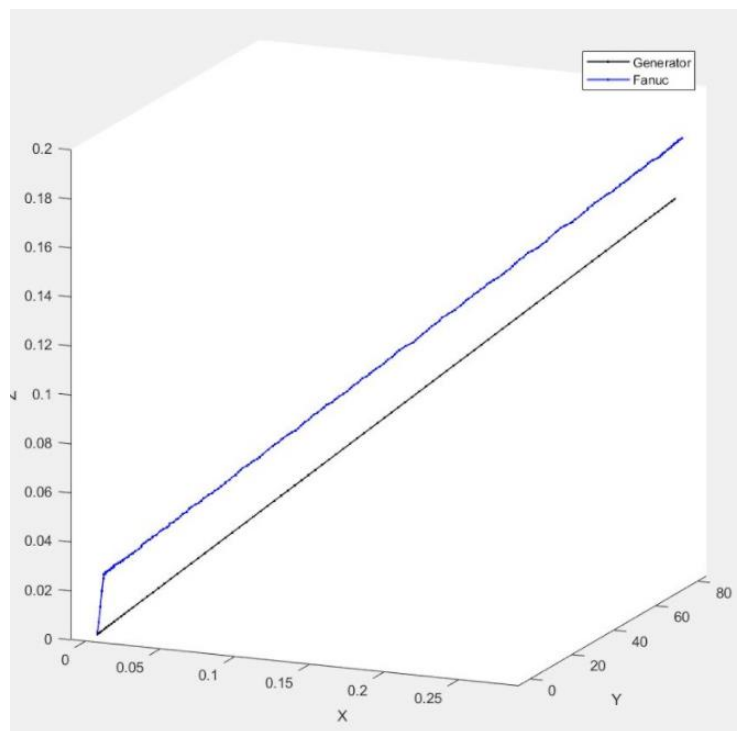
## Offline povelování robotu pomocí funkce SPLINE MOTION

Funkční balíček (Option) Spline Motion pro roboty Fanuc je především určen pro programování nestandardních pohybových trajektorií, kdy běžně používané pohybové příkazy Joint Motion (J), Linear Motion (L) nedostačují k aproximaci požadovaného komplexního pohybu robotu.

### Standardní pohybová funkce Move Linear (L)

I přesto byly však nejprve provedeny dílčí experimenty s těmito nativně v kontroleru robotu poskytovanými příkazy (J, L), neboť option Motion Spline je samostatně placeným doplňkem. Ve spolupráci s LT byl implementován postprocesor (Matlab), který generoval zájmové body robotu s

definovaným rozlišením, výstup postprocesoru byl zpracován samostatným programem (Basic) na tzv. LS soubor, který je standardním textově orientovaným vstupem kontrolerů robotů Fanuc a obsahuje všechny požadované informace (nastavení robotu, pohybové příkazy, zájmové body, atd.), v našem případě byl použit pohybový příkaz L (Linear Motion) s parametrem požadované translační rychlosti (v mm/s) a parametrem CNT 100 (tedy průjezd okolí bodu bez zastavení - tzv. blending). Záznam pohybu robotu byl zpracován přímo v kontroleru robotu v tzv. Background Tasku, viz [Background Task](#) a vyhodnocen opět v prostředí Matlab, kde byl porovnán s referenčním požadovaným pohybem. Výsledky jednoduchého přímkového pohybu s definovanou konstantní rychlostí (lichoběžníkový profil s omezeným zrychlením/zpomalením) jsou znázorněny na Obrázku 1.



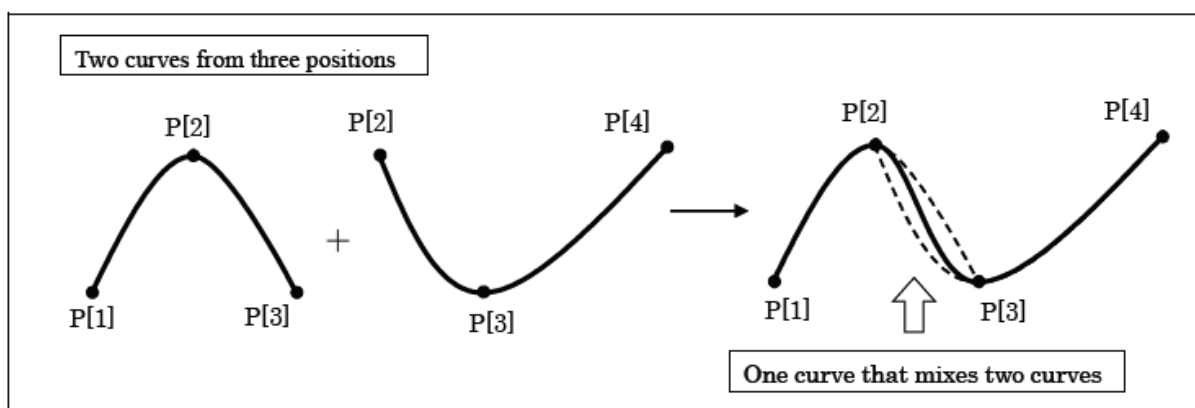
Obrázek 1: Přímkový pohyb robotu (pohybový příkaz L s definovanou rychlostí a blendingem)

Experimenty bohužel ukázaly, že přístup generování (aproximace) obecného pohybu prostřednictvím standardního pohybového příkazu L není zcela vhodné, a to z následujících důvodů:

- V podstatě není možné zjistit (nejlépe matematicky popsat), jak se chová funkce blending, která je parametrizována jediným parametrem CNT (0 - 100) - takové implementační poznámky Fanuc neposkytuje.
- Příkaz L není obecně vhodný (a ani určený) pro interpolaci "hustě" definovaných průjezdných bodů - nelze spolehlivě odhadnout výsledný pohyb ve smyslu jeho tvaru a rychlosti/zrychlení podél takové trajektorie.
- Z výše uvedených grafických výstupů je patrné, že ani jednoduchý lineární pohyb nelze uspokojivě rozdělit na dílčí pohyby s příkazy L - dochází k fluktuaci ve tvaru i rychlosti.
- Experimenty pro komplexnější trajektorie (dva lineární pohyby s polynomiálním napojením) byly provedeny se stejnými výsledky, proto je zde dále neuvádíme.

### Pohybová funkce Spline Motion (S)

Právě z výše uvedených důvodů bylo přistoupeno k využití placené option Spline Motion [3], která definuje nový pohybový příkaz S. Příkaz interpoluje trojici zájmových bodů blíže nespécifikovaným splinem. Pokud jsou interpolovány více než tři body, trajektorie je skládána ve smyslu kombinace vždy dvojice po sobě jdoucích spline interpolací, viz Obrázek 2. Opět bohužel platí, že bližší popis skutečného algoritmu výpočtu interpolační spline křivky není součástí uživatelské dokumentace.



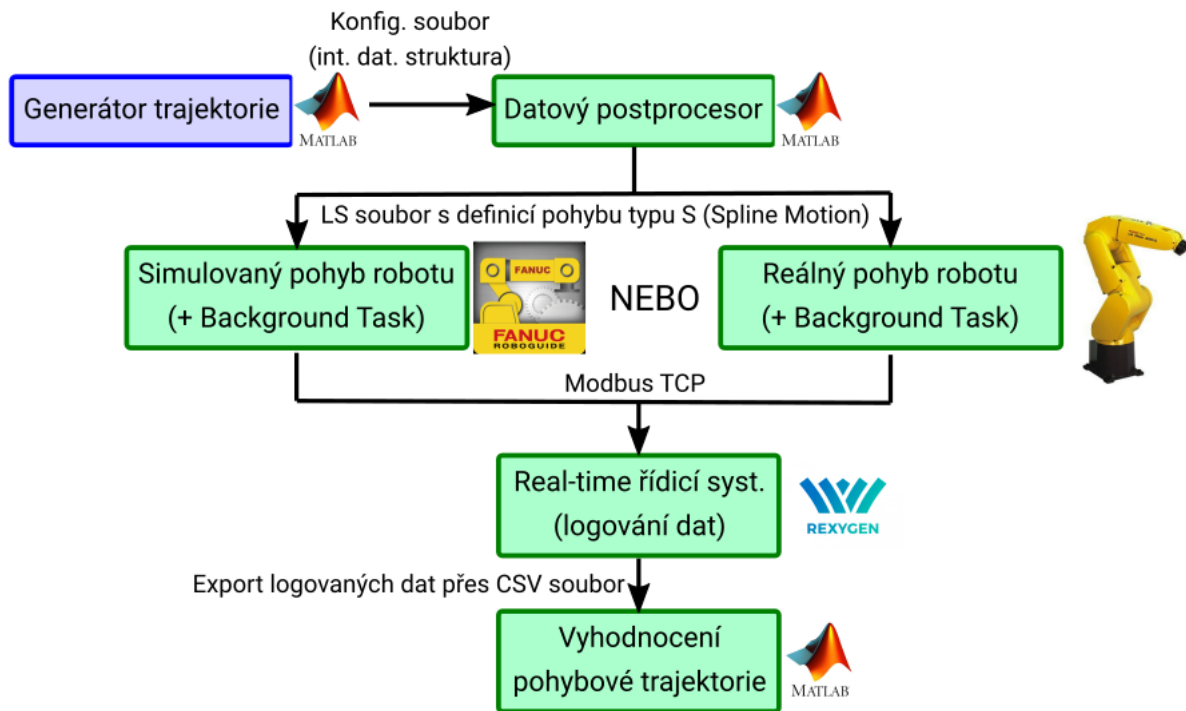
Obrázek 2: Uživatelský popis principu interpolace pohybů prostřednictvím option Spline Motion

Podobně jako v předchozím případě byl koncept experimentů s option Spline Motion založen na následujících technických prerekvizitách, které jsou podrobně popsány dále:

- **Datový postprocesor implementovaný** v prostředí Matlab, generující LS soubor s pohybovými příkazy S na základě poskytnutých dat z nadřazeného generátoru trajektorie (interní projekt ZČU, který není součástí řešených úloh v projektu)
- **Vlastní interpretace pohybu robotem** - simulace v programu RoboGuide, testy na reálném robotu
- **Zpracování zaznamenaných dat o výsledném pohybu robotu a komunikace zpět do prostředí Matlab pro vyhodnocení** založené na tzv. Background Tasku v kontroleru robotu a komunikačním protokolu Modbus TCP do real-time řídicího prostředí REXYGEN [4].
- **Vyhodnocení referenční a zaznamenané pohybové trajektorie v prostředí Matlab** - analýza přesnosti pohybu

Celkové přehledové schéma interfacu s robotem Fanuc založeném na využití option Spline Motion je

znázorněno na Obrázku 3.



Obrázek 3: Přehledové schéma interfacu s robotem Fanuc založeného na option Spline Motion (včetně vyhodnocení)

### Datový postprocesor

Datový postprocesor byl implementován v prostředí Matlab a generuje výstupní soubor s příponou LS, jako standardní skript pro programování robotů Fanuc prostřednictvím externích offline programovacích nástrojů (např. RoboDK, atd.). Struktura souboru LS je popsána na Obrázku 4.

Datový postprocesor využíval generátor trajektorie<sup>1</sup>, který ze zadaného typu pohybu (MoveJoint/MoveJointGroup, MoveLinear/MoveLinearGroup, MoveRecord) a parametrů generuje výstupní konfigurační soubor pro blok *RM\_Feed*<sup>2</sup>, který pak pohyb interpretuje v real-timeovém běhu řídicího systému REXYGEN. Tento přístup je v současnosti na ZČU používán pro robotické aplikace s komplexními pohyby, které je obtížné či nemožné implementovat přímo prostřednictvím funkčních bloků z knihovny Motion control řídicího systému REXYGEN. Právě výstupní konfigurační soubor a související datová struktura z generátoru trajektorie byla použita jako vstup datového postprocesoru.

<sup>1</sup>Generátor trajektorie není součástí řešení projektu. Jedná se o interní projekt ZČU (implementovaný v Matlabu), který je ve fázi výzkumu a vývoje, je stále rozšiřován a jeho použití je v současnosti umožněno pouze výzkumným pracovníkům.

<sup>2</sup>Funkční blok z knihovny Motion Control řídicího systému reálného času REXYGEN [4]  
[https://www.rexygen.com/doc/ENGLISH/MANUALS/BRef/RM\\_Feed.html](https://www.rexygen.com/doc/ENGLISH/MANUALS/BRef/RM_Feed.html)

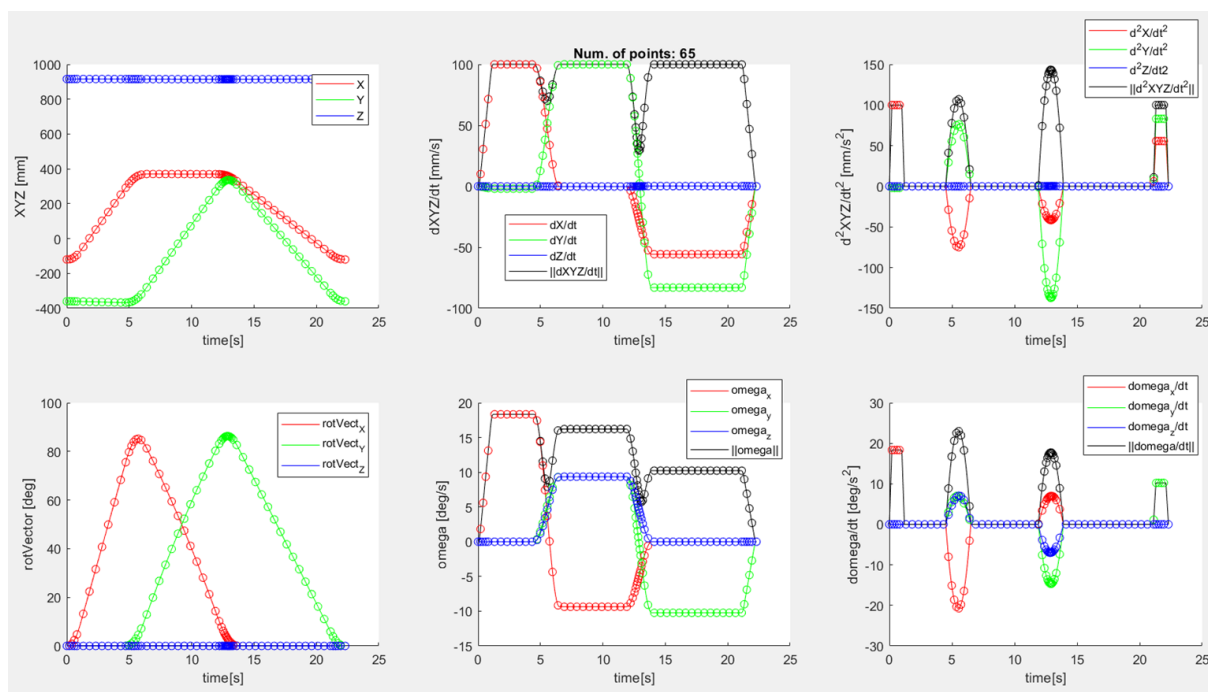
<pre> 1  /PROG FanucTest2_tdMax05tdMin005 2  /ATTR 3  COMMENT      = "FanucTest2_tdMax05tdMin005"; 4  /APPL 5  /MN 6 7  1: UTOOL_NUM=1; 8  2: UFRAME_NUM=0; 9  3:J P[1]  2% CNT100; 10 4:S P[2]  10mm/sec CNT100; 11 5:S P[3]  31mm/sec CNT100; 12 13 /POS 14 P[1]{ 15   GP1: 16   UF : 0, UT : 1, 17   J1= -108.435 deg, J2= -11.185 deg, J3=  5.382 deg, 18   J4=  0.000 deg, J5=  84.618 deg, J6= -288.435 deg 19 }; 20 P[2]{ 21   GP1: 22   UF : 0, UT : 1, 23   J1= -108.347 deg, J2= -11.231 deg, J3=  5.392 deg, 24   J4=  0.039 deg, J5=  84.492 deg, J6= -288.351 deg 25 }; 26 P[3]{ 27   GP1: 28   UF : 0, UT : 1, 29   J1= -107.795 deg, J2= -11.516 deg, J3=  5.452 deg, 30   J4=  0.272 deg, J5=  83.706 deg, J6= -287.822 deg 31 }; 32 P[4]{ 33   GP1: 34   UF : 0, UT : 1, 35   J1= -106.663 deg, J2= -12.074 deg, J3=  5.581 deg, 36   J4=  0.700 deg, J5=  82.104 deg, J6= -286.745 deg 37 }; 38 P[5]{ 39   GP1: 40   UF : 0, UT : 1, 41   J1= -104.913 deg, J2= -12.874 deg, J3=  5.790 deg, 42   J4=  1.234 deg, J5=  79.648 deg, J6= -285.086 deg 43 }; 44 /END </pre>	<p>Hlavička souboru (konfigurace)</p> <p>Pohybové příkazy a jejich parametry</p> <p>Definice zájmových bodů (v ACS)</p>
--	---

Obrázek 4: LS soubor s definicí pohybů a zájmových (průjezdných) bodů - zadáno v kloubových souřadnicích (ACS - Axes Coordinate System) jako výstup postprocesoru dat

Datový postprocesor je možné konfigurovat do několika režimů interpolace (převzorokování) dat z generátoru trajektorie za účelem generování LS souboru (toto nastavení ovlivňuje počet zájmových bodů v LS souboru a tím i počet příkazů S, neboť výstupem generátoru trajektorie je časová závislost poloh kloubových souřadnic robotu (ACS) s fixní periodou vzorkování):

- Vzorkování po konstantní ujeté vzdálenosti v translaci (v mm) či orientaci (v deg)
- Vzorkování po konstantním čase (v sekundách)

- Vzorkování po dynamicky měnících se časových okamžicích (v sekundách) z minimální do maximální nastavené hodnoty podle aktuální hodnoty zrychlení (v translaci nebo rotaci). Tento režim se zdá být nejvhodnější vzhledem k přesnosti sledování referenční trajektorie, neboť ve vysoce dynamicky se měnících úsecích trajektorie je nutné vzorkování pro option Spline Motion zjemňovat. Obrázek 5 demonstruje dynamický režim vzorkování pro konkrétní případ pohybové trajektorie.



Obrázek 5: Dynamické vzorkování, plná čára znázorňuje výstup generátoru trajektorie, vyznačené body pak vzorky reprezentující zájmové body v MCS (Machine Coordinate System), které budou po přepočtení do ACS tvořit průjezdné body pohybu S funkce Spline Motion

## Vlastní interpretace pohybu robotem

Vygenerovaný LS soubor byl využit k definici pohybů robotu Fanuc a to ve dvou variantách:

1. *Simulace pohybu robotu v prostředí RoboGuide<sup>3</sup>*, které umožňuje konfigurovat libovolný robot z portfolia FANUC a použít jej k simulaci pohybu robotu a okolní scény. Přesto, že simulovaný pohyb robotu poměrně přesně odpovídá reálnému robotu, nelze zde zcela zaručit striktní real-time běh simulátoru (lze jen monitorovat běh programu dle ukazatele v % ve smyslu 100% - simulace odpovídá realitě, < 100% simulace se zpožďuje, > 100% simulace se předbíhá). Tato vlastnost simulovaného režimu měla vliv na vyhodnocení přesnosti sledování pohybu robotem, neboť některá komunikovaná data byla v Background Tasku v RoboGuide předzpracována (výpočet rychlostí pohybu), viz dále.
2. *Testy na reálném robotu*, ve spolupráci s LT bylo provedeno testování na robotu Fanuc LR Mate 200iD. Výsledky odpovídaly simulacím, k problému s reálným během nedocházelo.

## Background Task

V simulovaném prostředí RoboGuide, stejně tak jako v reálném kontroleru robotu, lze využít tzv.

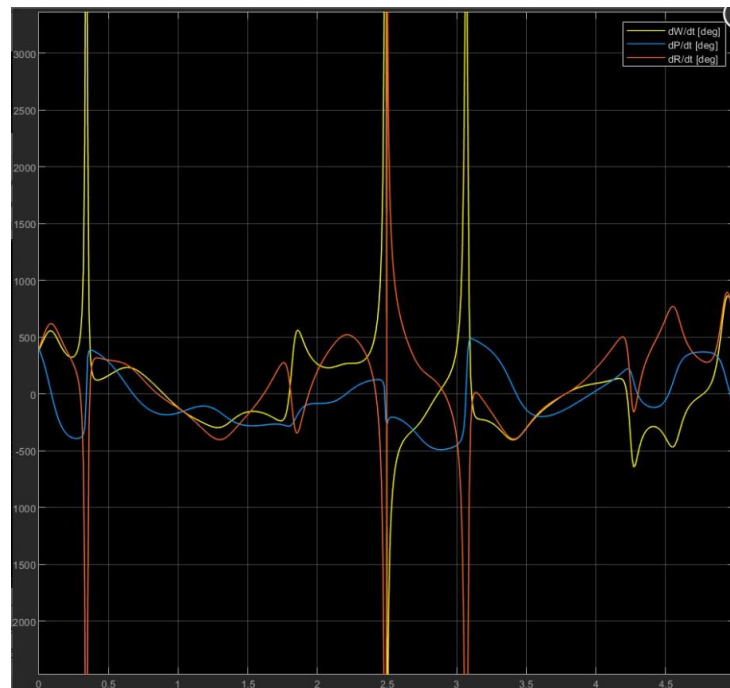
<sup>3</sup>ROBUGUIDE - Offline simulace pohybu a příkazů pomocí softwaru

<https://www.fanuc.eu/cz/cs/roboty/p%C5%99%C3%ADsllu%C5%A1enstv%C3%AD/roboguide>



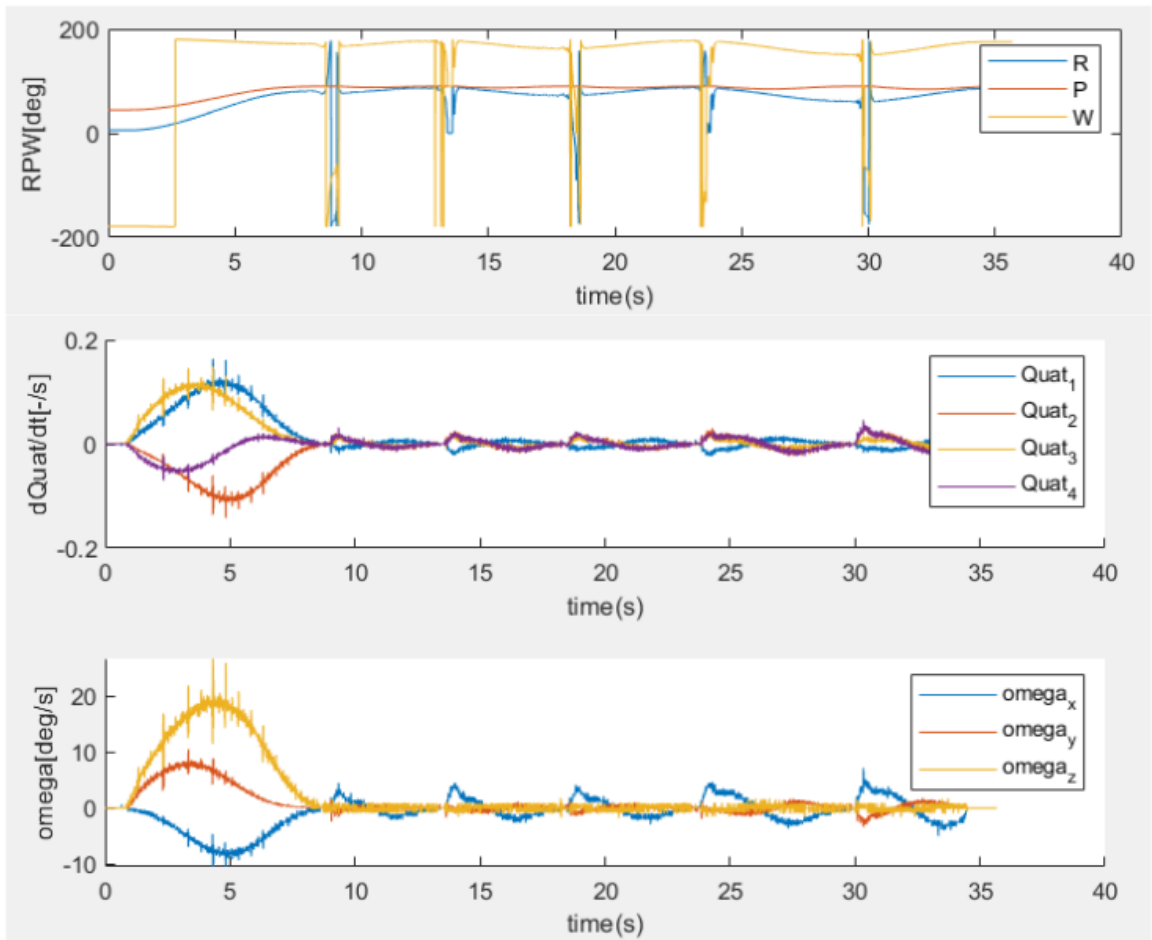
Background Task, viz Obrázek 8, který na pozadí běhu algoritmu řízení pohybu robotu může provádět dílčí operace nad Group IO registry. Vzhledem k faktu, že kontrolery robotu Fanuc standardně poskytují informace pouze o poloze (XYZ translace a WPR orientace<sup>4</sup>) a rychlosti TCP (rychlost TCP - jedna skalární hodnota pouze pro translaci), bylo nutné realizovat výpočet vektorové podoby rychlosti, tzn.:

- *Časové derivace polohy XYZ*: Odhad rychlosti v každé ose vypočítán jednostrannou časovou diferencí polohy.
- *Časové derivace orientace WPR*: Diferencování Eulerových úhlů obecně přináší komplikace z důvodu tzv. reprezentačních singularit, kdy pro specifické orientace dochází k přepínání mezi dvěma řešeními reprezentace prostřednictvím Eulerových úhlů (typicky pro úhel P v hodnotě +/- 90 deg.), viz Obrázek 6. Z tohoto důvodu byl realizován algoritmus přepočtu WPR úhlů na kvaterniony a časové diference byly provedeny na složkách kvaternionu. Zároveň lze prostřednictvím tzv. Eulerových kinematických rovnic stanovit z kvaternionu a jeho časové derivace vektor úhlové rychlosti). Reálné průběhy orientace WPR z robotu a odhad rychlostí je znázorněn na Obrázku 7.



Obrázek 6: Simulované průběhy časové derivace (rychlosti) Eulerových úhlů (fixní XYZ) - "úskoky" do vysokých (nekonečných) hodnot jsou způsobeny pohybem okolo singularity.

<sup>4</sup>WPR označují Eulerovy úhly ve schématu rotací XYZ okolo fixních os souřadného systému.



Obrázek 7: Odhad rychlostí orientace prostřednictvím časové derivace kvaternionu

```

1: ! Sign of life counter ;
2: R[1:Life counter]=(R[1:Life counter]+1) ;
3: ;
4: ! Actual Override ;
5: R[2:Override]=($MCR.$GENOVERRIDE) ;
6: ;
7: ! Cartesian pos WORLD ;
8: R[3:POS_X]=($SCR_GRP[1].$MCH_POS_X) ;
9: R[4:POS_Y]=($SCR_GRP[1].$MCH_POS_Y) ;
10: R[5:POS_Z]=($SCR_GRP[1].$MCH_POS_Z) ;
11: R[6:POS_W]=($SCR_GRP[1].$MCH_POS_W) ;
12: R[7:POS_P]=($SCR_GRP[1].$MCH_POS_P) ;
13: R[8:POS_R]=($SCR_GRP[1].$MCH_POS_R) ;
14: ;
15: !Quaternions ;
16: R[31:sy]=R[8:POS_R]*.5 ;
17: R[30:cy]=(COS[R[31]]) ;
18: R[31:sy]=(SIN[R[31]]) ;
19: R[33:sp]=R[7:POS_P]*.5 ;
20: R[32:cp]=(COS[R[33]]) ;
21: R[33:sp]=(SIN[R[33]]) ;
22: R[35:sr]=R[6:POS_W]*.5 ;
23: R[34:cr]=(COS[R[35]]) ;
24: R[35:sr]=(SIN[R[35]]) ;
25: R[9:Q_1]=(R[34:cr]*R[32:cp]*R[30:cy]+R[35:sr]*R[33:sp]*R[31:sy]) ;
26: R[10:Q_2]=(R[35:sr]*R[32:cp]*R[30:cy]-R[34:cr]*R[33:sp]*R[31:sy]) ;
27: R[11:Q_3]=(R[34:cr]*R[33:sp]*R[30:cy]+R[35:sr]*R[32:cp]*R[31:sy]) ;
28: R[12:Q_4]=(R[34:cr]*R[32:cp]*R[31:sy]-R[35:sr]*R[33:sp]*R[30:cy]) ;
29: ;
30: !Q1 > 0 ;
31: IF (R[9:Q_1]<0) THEN ;
32: R[9:Q_1]=(0-R[9:Q_1]) ;
33: R[10:Q_2]=(0-R[10:Q_2]) ;
34: R[11:Q_3]=(0-R[11:Q_3]) ;
35: R[12:Q_4]=(0-R[12:Q_4]) ;
36: ENDIF ;
37: ;
38: ! Cartesian axes speed calc ;
39: R[13:SPD_X]=(($SCR_GRP[1].$MCH_POS_X-R[36:POS_PREV_X])/.008) ;
40: R[14:SPD_Y]=(($SCR_GRP[1].$MCH_POS_Y-R[37:POS_PREV_Y])/.008) ;
41: R[15:SPD_Z]=(($SCR_GRP[1].$MCH_POS_Z-R[38:POS_PREV_Z])/.008) ;
42: ;
43: !Quaternions difference calc ;
44: R[16:SPD_Q1]=((R[9:Q_1]-R[39:POS_PREV_Q1])/.008) ;
45: R[17:SPD_Q2]=((R[10:Q_2]-R[40:POS_PREV_Q2])/.008) ;
46: R[18:SPD_Q3]=((R[11:Q_3]-R[41:POS_PREV_Q3])/.008) ;
47: R[19:SPD_Q4]=((R[12:Q_4]-R[42:POS_PREV_Q4])/.008) ;
48: ;
49: ! Store positions for speed calc ;
50: R[36:POS_PREV_X]=($SCR_GRP[1].$MCH_POS_X) ;
51: R[37:POS_PREV_Y]=($SCR_GRP[1].$MCH_POS_Y) ;
52: R[38:POS_PREV_Z]=($SCR_GRP[1].$MCH_POS_Z) ;
53: R[39:POS_PREV_Q1]=(R[9:Q_1]) ;
54: R[40:POS_PREV_Q2]=(R[10:Q_2]) ;
55: R[41:POS_PREV_Q3]=(R[11:Q_3]) ;
56: R[42:POS_PREV_Q4]=(R[12:Q_4]) ;

```

Obrázek 8: Zdrojový kód Background tasku

### Komunikační rozhraní do systému REXYGEN

Komunikace dat z Background Tasku kontroleru robotu Fanuc nebo simulovaného prostředí RoboGuide do řídicího systému REXYGEN za účelem jejich logování bylo zajištěno přes průmyslový komunikační protokol Modbus TCP. Konkrétně byla ke komunikaci ze strany kontroleru využita funkce *HMI DEVICE COMMUNICATION*, která je součástí standardního vybavení *Standard setting (R651)* a

měla by být tedy k dispozici ve všech standardních kontrolerech Fanuc. Funkce je klasicky využívána pro tvorbu HMI na externích zařízeních a umožňuje zápis i čtení hodnot o stavu robotu pomocí protokolů Modbus TCP/RTU nebo OPC UA. Detailní popis funkce včetně postupu jejího nastavení lze nalézt v [3] na straně 518.

*HMI DEVICE COMMUNICATION* má předpřipravený Modbus registr, kam jsou namapovány digitální, analogové, uživatelské, skupinové (group) vstupy a výstupy (DI, DO, AI, AO, UI, UO, GI, GO), numerické (R), polohové registry (PR) a další proměnné. Nastavení tohoto Modbus registru včetně adres a typu hodnot je vidět v Tabulce 1.

Table	Address range	Corresponded robot data (a: address)	
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	Not used	
Coils	1 ~ 10000	Digital output	DO[a]
	10001 ~ 20000	Robot output	RO[a-10000]
	20001 ~ 30000	Flag	F[a-20000]
	30001 ~ 65536	Not used	
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	Not used	
Holding Registers	1 ~ 16384	Robot data is assigned (assigned to R[a] by default)	
	16385 ~ 16448	55.4 DYNAMIC ASSIGNMENT TO HOLDING REGISTERS using this area.	
	16449 ~ 65536	Not used	

Tabulka 1: Modbus registr poskytující informace o aktuálním stavu robotu ve funkci HMI Device Communication

V Background tasku jsou sledované hodnoty ukládány do numerických registrů R[1]-R[19] a pro jejich posílání je tedy vhodné využít *Holding Registers*. Hodnoty byly přiřazeny registrům podle Tabulky 2.

Name	Fanuc		Rexygen	
	Register	Address	Address	Type
Life counter	1	1-2	0-1	INT32
Override	2	3	2	INT16
POS_X	3	4-5	3-4	FLOAT
POS_Y	4	6-7	5-6	FLOAT
POS_Z	5	8-9	7-8	FLOAT
POS_W	6	10-11	9-10	FLOAT
POS_P	7	12-13	11-12	FLOAT
POS_R	8	14-15	13-14	FLOAT
Q_1	9	16-17	15-16	FLOAT
Q_2	10	18-19	17-18	FLOAT
Q_3	11	20-21	19-20	FLOAT
Q_4	12	22-23	21-22	FLOAT
SPD_X	13	24-25	23-24	FLOAT
SPD_Y	14	26-27	25-26	FLOAT
SPD_Z	15	28-29	27-28	FLOAT
SPD_Q1	16	30-31	29-30	FLOAT
SPD_Q2	17	32-33	31-32	FLOAT
SPD_Q3	18	34-35	33-34	FLOAT
SPD_Q4	19	36-37	35-36	FLOAT

Tabulka 2: Seznam posílaných hodnot pomocí Modbus TCP

Tabulka udává v prvním sloupci jméno komunikované proměnné, ve druhém číslo přiřazeného registru R, ve třetím Modbus adresu ve Fanuc, ve čtvrtém adresu v REXYGENU a v posledním sloupci typ proměnné v prostředí REXYGEN.

### Vyhodnocení pohybové trajektorie

Vyhodnocení pohybové trajektorie bylo provedeno v prostředí Matlab a jeho cílem bylo porovnat zaznamenaná data z robotu či simulace s referenční generovanou trajektorií pohybu. Důraz byl kladen především na:

- Absolutní odchylky polohy v translaci a orientaci
- Absolutní odchylky rychlosti (požadovaného profilu rychlosti) v translaci a orientaci

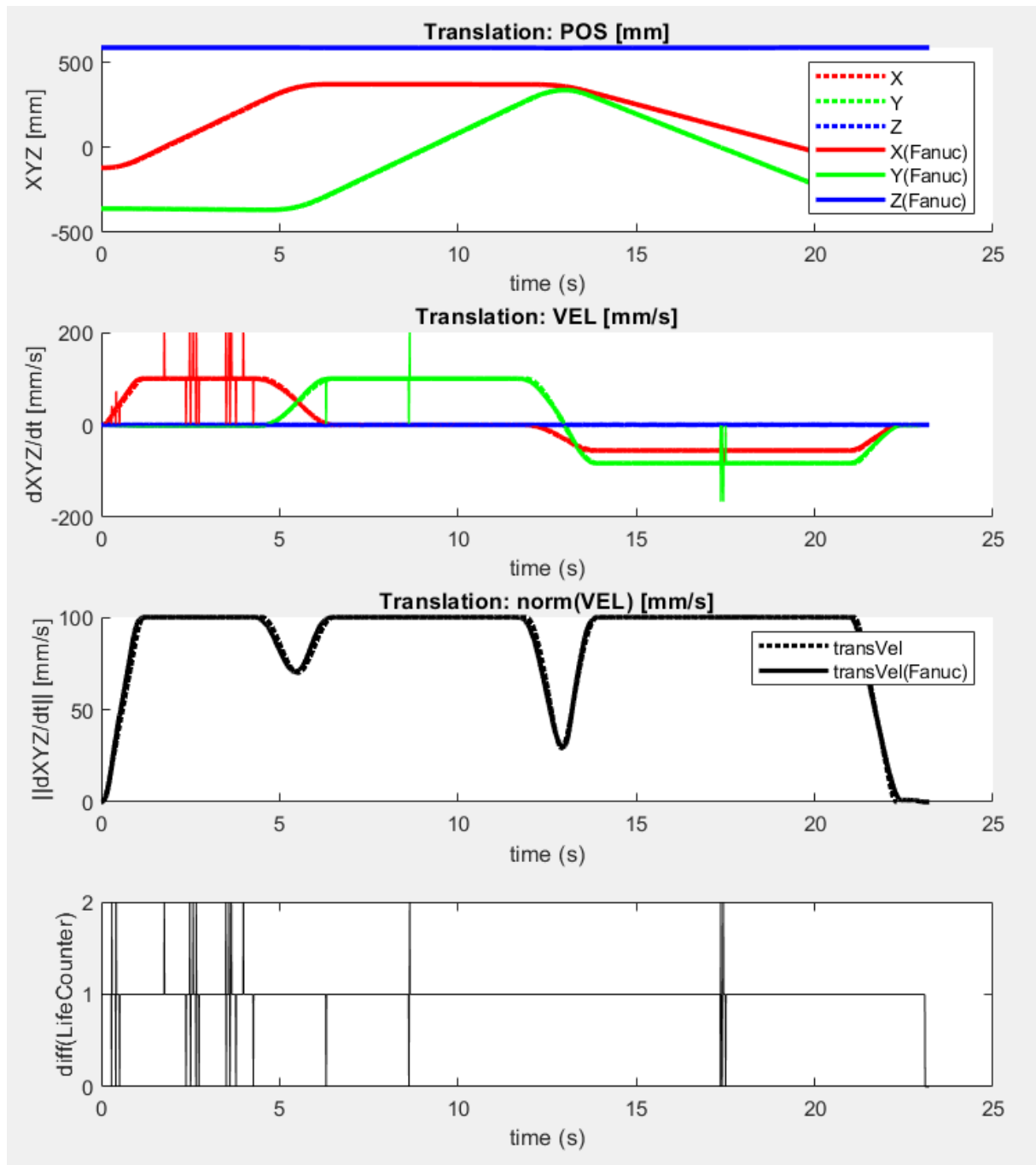
Obrázek 9 ukazuje polohy a rychlosti robotu v translaci v souřadnicích XYZ (včetně normy rychlosti reprezentující profil translační rychlosti podél trajektorie pohybu). Tenčí čarou v druhém grafu je reprezentována diference (odhad časové derivace - rychlosti) z polohových dat XYZ v prostředí Matlab (tlustá čára v druhém grafu jako taková reprezentuje XYZ rychlosti odhadnuté v Background Tasku a komunikované do REXYGENU). Pozorovatelné "úskoky" souvisejí s problémy s během robotu v simulovaném prostředí RoboGuide, viz výše, a korespondují s odchylkou diference pomocné komunikované hodnoty LifeCounter od hodnoty 1. LifeCounter reprezentuje celočíselnou hodnotu, která je v každé periodě (zde 8 ms) běhu Background Tasku inkrementována o 1. Vzhledem k tomu, že perioda logovacího tasku v systému REXYGEN je také 8 ms, měla by se logovaná hodnota LifeCounter zvýšit v každé periodě právě o 1. Pokud nedojde ke zvýšení o 1 nebo naopak dojde ke zvýšení o více než jedna, znamená to, že se simulované prostředí RoboGuide zpomaluje nebo předbíhá v porovnání s real-time během. Tento jev nenastává při běhu Background Tasku na reálném robotu.

Obrázek 10 ukazuje odchylky polohy a rychlosti robotu v souřadnicích XYZ (včetně normy rychlosti) mezi referenční plánovanou pohybovou trajektorií a logovanou reálně robotem vykonanou pohybovou trajektorií.

Obrázek 11 ukazuje polohy a rychlosti robotu v orientaci v kvaternionech (včetně vektoru úhlové

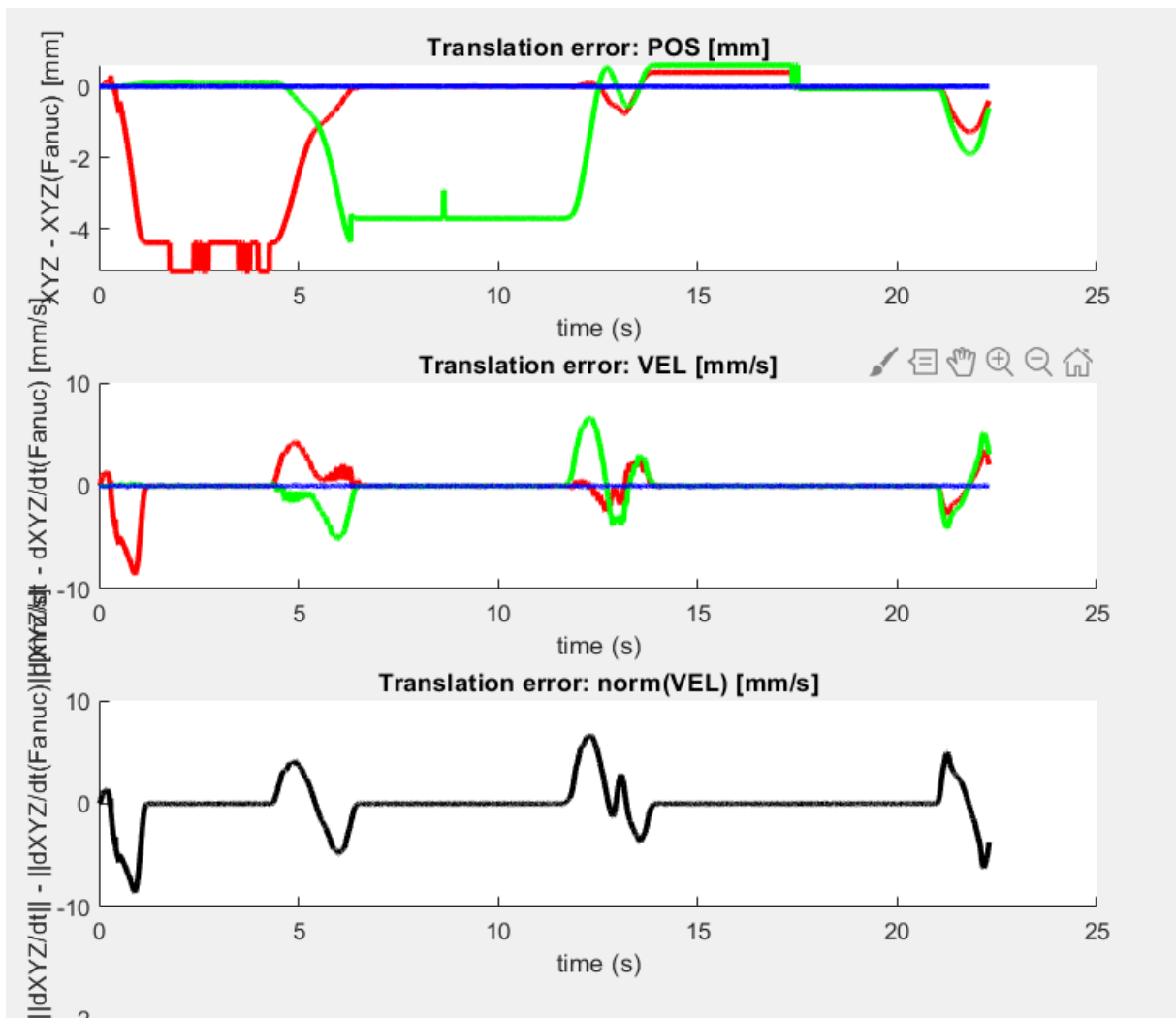
rychlosti a jeho normy reprezentující profil úhlové rychlosti podél trajektorie pohybu).

Obrázek 12 ukazuje odchylky polohy robotu jako rotační vektor<sup>5</sup> a odchylku úhlové rychlosti robotu (včetně normy úhlové rychlosti).

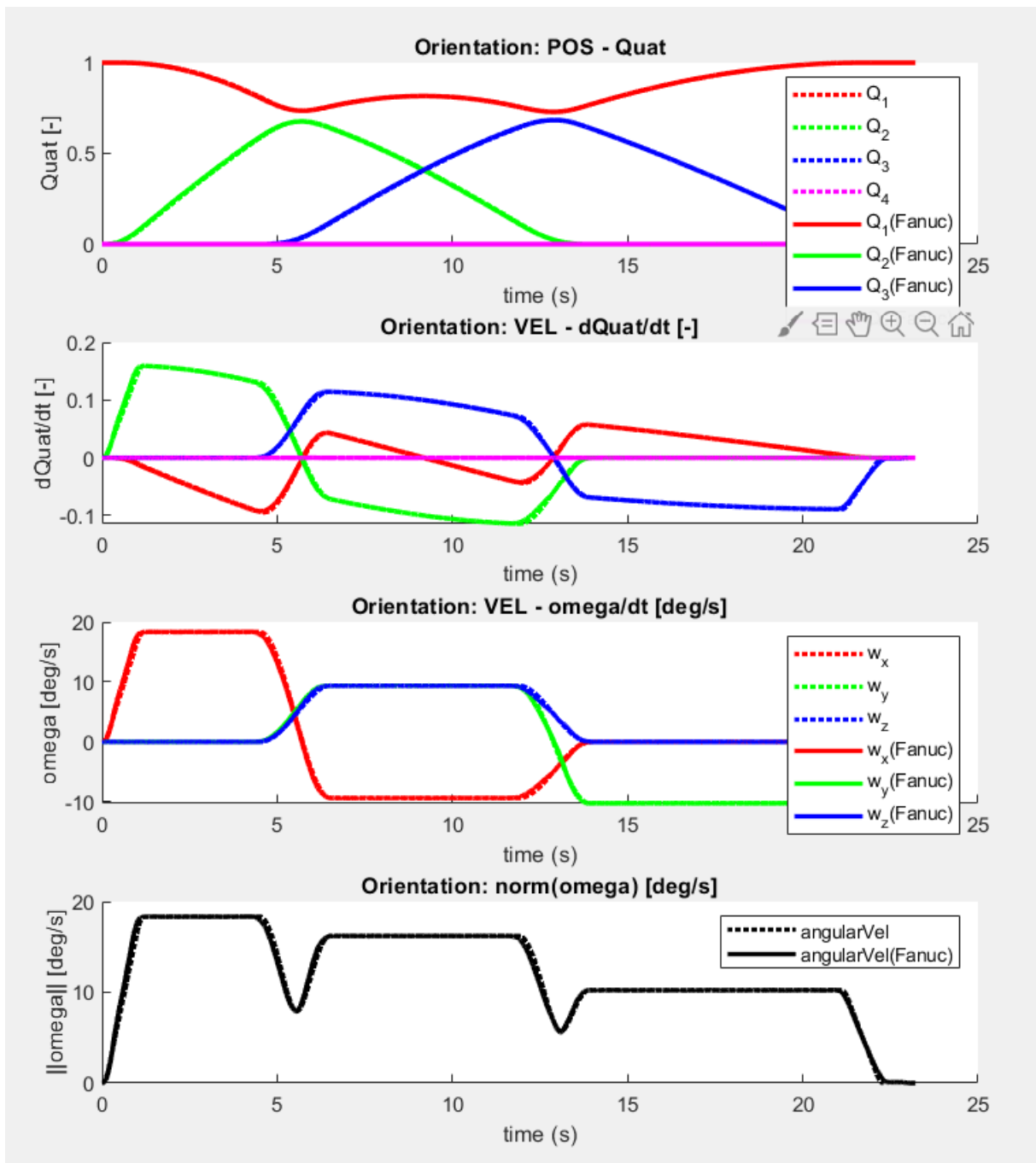


Obrázek 9: Porovnání plánované a skutečné pohybové trajektorie - translace

<sup>5</sup>Rotační vektor je vektor, jehož směr určuje aktuální osu rotace a jeho velikost určuje úhel otočení kolem této osy (takovou reprezentaci pohybu používají např. roboty UR).

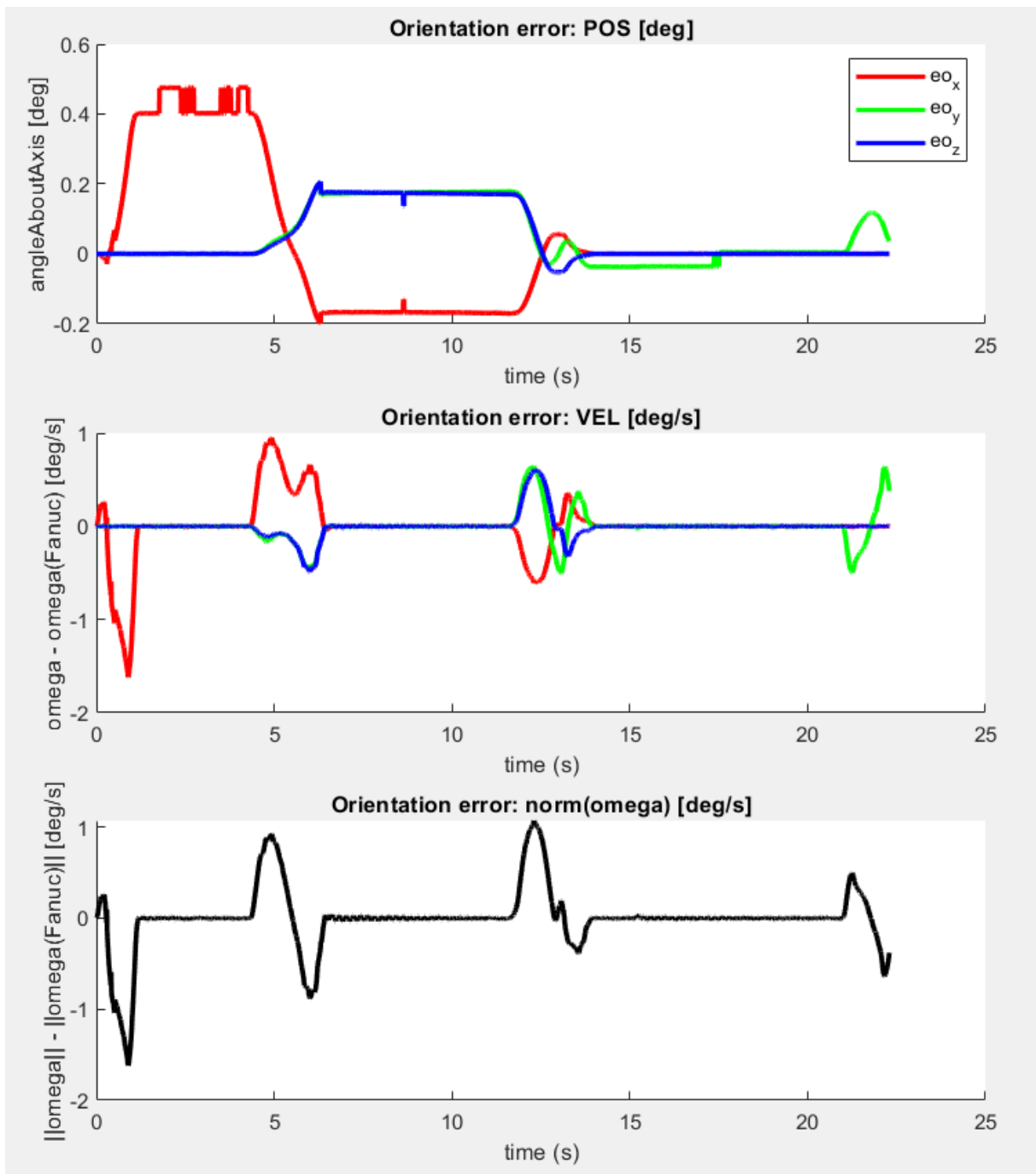


Obrázek 10: Odchyly mezi plánovanou a skutečnou pohybovou trajektorií - translace



Obrázek 11: Porovnání plánované a skutečné pohybové trajektorie - orientace





Obrázek 12: Odchylky mezi plánovanou a skutečnou pohybovou trajektorií - orientace

## Online povelování robotu pomocí funkce DYNAMIC PATH MODIFIER

Funkční balíček (Option) Dynamic Path Modifier (DPM) pro roboty Fanuc je určen především ke korekci předem naprogramovaných pohybů v reálném čase na základě vnějších snímačů. Při vhodném nastavení jej lze s nijak velkým omezením použít i pro řízení polohy všech os robotu na základě dat komunikovaných z vnějšího řídicího systému v reálném čase.

Option DPM je možné pro podporované kontrolery u výrobce standardně objednat bez dalších omezení.

Možnosti funkce DPM lze shrnout následovně:

- Funkce je aktivována ze standardního TP programu instrukcí "Track".
- DPM lze rozsáhle parametrizovat. Je možné vytvořit jeden nebo více tzv. "schedule" a v instrukci "Track" se na ně odkazovat.
- DPM pracuje ve dvou základních režimech - Modal nebo Inline. Režim Modal provádí korekci trajektorie robotu v reálném čase, se základní časovou periodou interpolace (ITP, zpravidla 8ms). Režim Inline provádí korekci pouze cílové polohy jednotlivých instrukcí, a pro uvažované aplikace není zajímavý.
- V rámci parametrizace lze volit:
  - souřadný systém, ve kterém je korekce aplikována (user frame nebo user tool),
  - osy, pro které je korekce povolena - lze zapnout jednotlivě pro všechny translační (x, y, z) i rotační (w, p, r),
  - pro každou osu maximální celkovou velikost korekce polohy od bodu, kde bylo DPM aktivováno,
  - pro každou osu maximální velikost korekce polohy za jednu periodu ITP - tj. ve výsledku maximální rychlost,
  - pro každou osu zdroj hodnoty korekce a její škálování - lze použít analogové vstupy, systémové proměnné a Group registry,
  - podporu režimu "Stationary track" - pro uvažovaný způsob použití velmi důležité, umožňuje aplikovat vnější korekci polohy i po dokončení pohybu z vlastního TP programu, dále se nastavuje DI signál, kterým je tento režim řízen.
- Z pohledu návaznosti na vnější řídicí systém se nabízí využití tzv. Group registrů, do kterých je možné namapovat hodnoty z libovolného robotem podporovaného a nainstalovaného komunikačního protokolu. Určitým omezením je, že Group registry mají velikost pouze 16 bitů, a tomu je třeba přizpůsobit nastavení škálování. S ohledem na dále uvedené to ale v praxi nepředstavuje problém.
- Vstupní hodnoty pro DPM představují inkrementální korekci polohy za časovou periodu ITP, nikoliv korekci absolutní. Z pohledu vnějšího systému je tedy interpretace hodnot směrem do DPM rychlost, nikoliv poloha. Určitou výhodou tohoto řešení je, že nevádí omezený dynamický rozsah Group registrů (16 bitů, viz výše) a že nulová hodnota je vždy "bezpečná".

Z pohledu integrace se standardním programovým prostředím robotu, tj. TP, se funkce DPM jeví velmi použitelně a přímočaře. Je možné snadno kombinovat pohyby generované standardním programem, v určitém bodě přejít na vnější řízení přes DPM a následně zpět.

```

DPM DET SCH
123456789012345678901234567890123456789012345678901234
SCHD 1, DPMGRP 1, CHAN 1      11/13

 1 DPM type:                MODAL
 2 Offset ref. frame:      UFRAME
 3 Offset accumulate:      FALSE
 4 Stationary Track:       NO
 5 Station Track sych:     DI[ 0]
 6 Channel enabled:        TRUE
 7 On-The-Fly Input:       DI[ 0]
 8 Max offset/path:        10.00 mm
 9 Max offset/ITP:         0.20 mm
10 Min offset/ITP:         0.00 mm
11 Offset input type:      GROUP
12 Group Index:            GI[ 10]
13 Scale factor:           1000.00

SCHEDULE DPMGRP CHANNEL [CHOICE] HELP

```

Obrázek 13: Konfigurační dialog DPM Schedule v kontroleru Fanuc.

Pro ověření skutečných vlastností funkce DPM byla provedena řada testů na reálném robotu Fanuc LRmate s kontrolerem R-30iB Mate Plus.

Robot byl s vnějším řídicím systémem propojen real-time komunikačním protokolem EtherNet/IP se základním časovým cyklem (RPI) 8ms. V TP programu byl aktivován režim DPM ve všech osách robotu pomocí instrukce "Track". Pro kontrolu skutečného pohybu robotu byly aktuální polohy a rychlosti komunikovány zpět do řídicího systému přes Group IO registry s využitím výpočtů v tzv. Background Task, viz [Background Task](#).

**Bylo zjištěno, že zpoždění v celé smyčce (od nastavení požadavku na změnu polohy k dosažení této polohy) se pohybuje kolem 30-40 ms.** S ohledem na princip protokolu Ethernet/IP lze předpokládat, že 50-75% z tohoto času připadá na režii samotné komunikace.

**Dále bylo zjištěno, že sledování referenčních trajektorií je velmi přesné, bez neočekávaných efektů interpolace nebo jiných filtrů.**

S ohledem na skutečnost, že roboty jsou v praxi zpravidla nasazovány v kombinaci s nadřazeným řízením v podobě PLC automatů (Programmable Logic Controller), se řízení robotu s využitím funkce DPM jeví jako reálně použitelná varianta. Moderní PLC umožňují dostatečně rychlou komunikaci a disponují dostatečným výpočetním výkonem a paměťovou kapacitou, aby mohly být pro reprodukci nebo generování pohybové trajektorie pro funkci DPM použity. S roboty Fanuc je možné komunikovat i z více vnějších systémů najednou pomocí více různých protokolů, takže i v případě použití PLC s nedostatečným výkonem je možné pro generování DPM použít např. další průmyslový počítač.

Další aktivita by měla směřovat k ověření použitelnosti dalších síťových komunikačních protokolů z pohledu real-time vlastností, zejména pak protokolu Modbus TCP.

Závěrem lze uvést, že funkce DPM je dobře použitelná pro "on-line" řízení robotu, nevykazuje žádné skryté nedokumentované chování, a případně umožňuje obejít i některé zjištěné nedostatky funkce SPLINE MOTION, jak je popsáno v předchozí kapitole. Nevýhodou může být nutnost implementace generátoru trajektorie v externím systému (např. PLC) se schopností real-time komunikace s robotem.

## Zhodnocení získaných výsledků

Software interfacu s průmyslovým robotem se zaměřením na výrobce Fanuc byl realizován s ohledem na předpokládanou finální pilotní aplikaci ve variantě pomocí funkce SPLINE MOTION, viz [Offline povelování robotu pomocí funkce SPLINE MOTION](#). Dosažené výsledky lze shrnout následovně:

- SW komunikace s roboty Fanuc je určen pro offline povelování robotu, tzn. není určen pro zpětnovazební řízení, což vyhovuje podstatě pilotní aplikace
- SW je realizován jako skript v prostředí Matlab (datový postprocesor) + konfigurace kontroleru robotu + SW pro vyhodnocení (konfigurace v řídicím SW REXYGEN + vyhodnocovací skript v Matlabu) - samostatně lze SW využít pro offline povelování robotů Fanuc s option Spline Motion
- Přesnost sledování referenční trajektorie je v případě pohybů cca 100 mm/s a 20 deg/s se zrychleními 125 mm/s<sup>2</sup> a 25 deg/s<sup>2</sup> přibližně 4mm v translaci a 0.5 deg v orientaci. Sledování rychlosti se pohybuje přibližně s přesností 8 mm/s v translaci a 1.5 deg/s v orientaci.
- SW lze alternativně rozšířit pro online povelování robotů Fanuc prostřednictvím option Dynamic Path Modifier, viz [Online povelování robotu pomocí funkce DYNAMIC PATH MODIFIER](#) - takový přístup je potenciálně možné aplikovat na roboty i jiných výrobců, kteří podporují analogické možnosti interakce s nadřazeným řídicím systémem.

## Dostupnost SW

Finální software v podobě zdrojových kódů bude s ohledem na stále probíhající dílčí činnosti (opravy, upravy, atd.) kompletně k dispozici ke stažení po ukončení projektu na zveřejněném odkazu.

V současné podobě je SW ke stažení na odkazu uvedeném v protokolu o testování SW:

*[O] Návrh interfacu s průmyslovým robotem (Protokol o testování SW)*

SW je součástí hlavního výstupu projektu:

- [Gprot] Prototyp záznamového zařízení (ukončení 8/2021)
- [Gprot] Prototyp robotické buňky osazené záznamovým zařízením (ukončení 8/2021)

## Reference

- [1] J. Reitinger, V. Šetka: Aktivita 3.3: Vývoj interfacu s průmyslovým robotem, Západočeská univerzita v Plzni, 2019.
- [2] Software interfacu s průmyslovým robotem (Protokol o testování SW), Západočeská univerzita v Plzni, 2020.
- [3] FANUC Robot series Optional Function OPERATOR'S MANUAL, B-83284EN-2/09, FANUC CORPORATION, 2012.
- [4] REXYGEN: Pokročilý nástroj pro řízení strojů a procesů, <https://www.rexygen.com/cs/>