

TAČR Centrum kompetence CIDAM

Motion control software for robotic manipulators operating in
industrial parts cleaning applications

Výzkumná zpráva WP5-DV091

(User documentation of developed motion control software: specification of programming, description
of human-machine interface)

Martin Švejda, Martin Goubelj, Arnold Jáger

3. ledna 2019

T A
Č R



Úvod

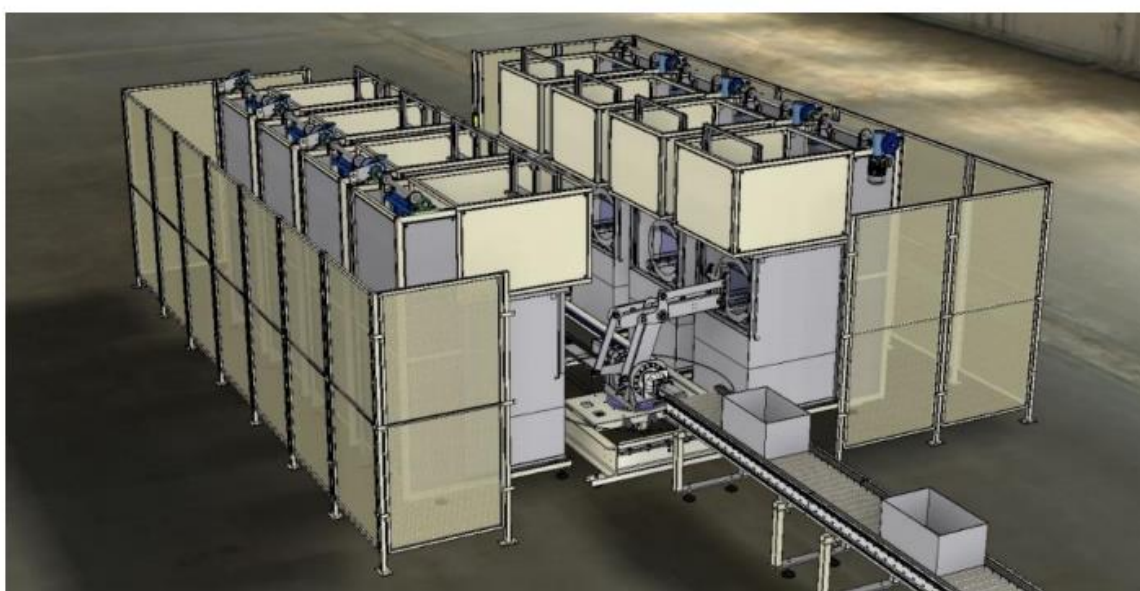
Předložená technická zpráva reprezentuje dokumentaci softwaru pro řízení pohybu robotu typu Zakladač a Vodník, viz Obrázek 1, vyvinutého v rámci projektu. Dokumentace popisuje jednotlivé vrstvy řídicího systému z hlediska použití uživatelem a některé rámcové implementační detaily. U dílčích skupin funkčních bloků je uveden výčet vstupů, výstupů a parametrů. Detailní uživatelská dokumentace je součástí zdrojů uvedených ve výčtu referencí.

Řídicí systém robotů je implementován dle následujících pravidel:

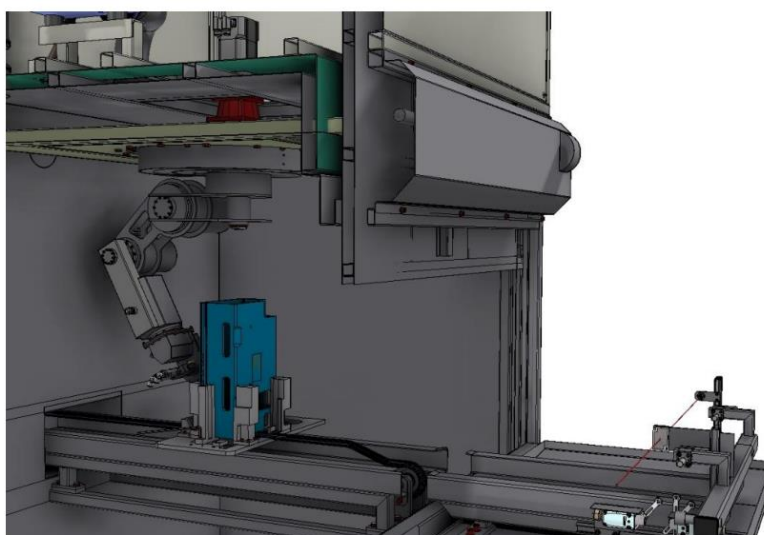
- Real-Time jádro řídicího systému pohybu implementováno v nástroji REXYGEN [1]
 - Knihovny funkčních bloků pro procesní řízení (PI regulace, pokročilé regulátory, atd.)
 - Knihovny funkčních bloků pro zpracování signálů (filtry, selektory, atd.)
 - Knihovny funkčních bloků pro řízení pohybu (dle normy PLCopen Motion Control [2])
 - Rozhraní pro průmyslové komunikace (CAN, EtherCAT, Modbus, atd.)
 - Diagnostika
- Řízení samostatných pohonů (motorů) kloubů robotu zajištěno servoměniči (TGdrives, Omron) s vlastním firmwarovým vybavením (parametrizovatelná kaskádní regulace servomotorů), povelování servoměničů přes průmyslovou komunikaci EtherCAT
- Generování pohybových trajektorií robotů
 - Standardní pohyby (např. osový JOG, jednoduchý koordinovaný JOG) implementovány přímo funkčními bloky z knihovny REXYGEN
 - Pokročilé plánování pohybu realizováno mimo jádro řídicího systému v podobě zpracování oddělenou aplikací (editorem) implementovaným v prostředí Matlab
- Virtuální simulační modely (simulace generovaných pohybových trajektorií) implementovány v prostředí Matlab/Simulink/SimMechanics
- Vizualizace/operátorské rozhraní implementováno v nástroji Inkscape (napojení ovládaných/zobrazovaných signálů do jádra řídicího systému, standardní možnosti systému REXYGEN včetně možností tvorby 3D vizualizací)
- Implementace algoritmů pro vyvíjené roboty, zejména pak implementace přímé a inverzní kinematiky, jsou integrovány do jádra řídicího systému prostřednictvím speciálních funkčních bloků (*MCP_SetKinTransform*) dle normy PLCopen Motion Control. Implementace vychází z dokumentace [3, 4].



Obrázek 1: Robot typu Zakladač (vlevo) a Vodník (vpravo)



Obrázek 2: Vizualizace - integrace robotu Zakladač do technologického celku



Obrázek 3: Vizualizace - zástavba robotu Vodník do kabiny odmašťovacího stroje

Vrstvy řídicího systému

Navržený řídicí systém robotů je koncipován v několika hierarchických vrstvách, viz Obrázek 19, které jsou popsány dále:

Vrstva 0: Mechanika robotů

Mechanické konstrukční provedení robotů bylo zajištěno partnerem Eurotec JKR s.r.o. a dokumentováno v příslušných technických zprávách [5,6,7]. Byla vyvinuta produkční řada tří manipulátorů s paralelní kinematikou s různými specifikacemi nosnosti a pracovního prostoru pro obsluhu mycích kabin, interní označení Zakladač (viz Obrázek 1 vlevo). Druhým typem je integrovaný robot určený pro přímou zástavbu kabin mycích strojů, typ Vodník (Obrázek 1 vpravo), zpracovaný opět ve třech velikostních variantách. Obě typové řady jsou určeny pro použití v technologii průmyslového odmašťování a odlakování. Způsob integrace do technologického celku ukazuje 3D vizualizace na Obrázcích 2 a 3.

Vrstva 1: Výkonová elektronika (Servoměniče)

Řídicí elektronika pro servomotory v kloubech robotů byly realizovány standardními komerčními servoměniči (TG drives, Omron) s příslušným firmwarem. Komunikace s měniči byla realizována přes komunikační protokol EtherCAT přímo z jádra řídicího systému REXYGEN prostřednictvím příslušného ovladače (rozhraní).

Vrstva 2: Komunikace se servoměniči, Komunikace s periferiemi

Vrstva tvořená ovladači řídicího systému, kde je podpořena celá řada komunikačních protokolů: CAN, EtherCAT, Modbus, atd.

Komunikační vrstva slouží nejen ke komunikaci s jednotkami servoměničů, ale také s dalšími periferními zařízeními. Typicky se jedná např. o vstupně/výstupní karty průmyslových PLC (např. B&R) pro ovládání digitálních a analogových vstupů/výstupů, či např. komunikaci s diagnostickými zařízeními a dalšími periferiemi (displeje, atd.). Při vývoji této vrstvy byly využity výsledky dosažené v předchozích etapách projektu [8,9].

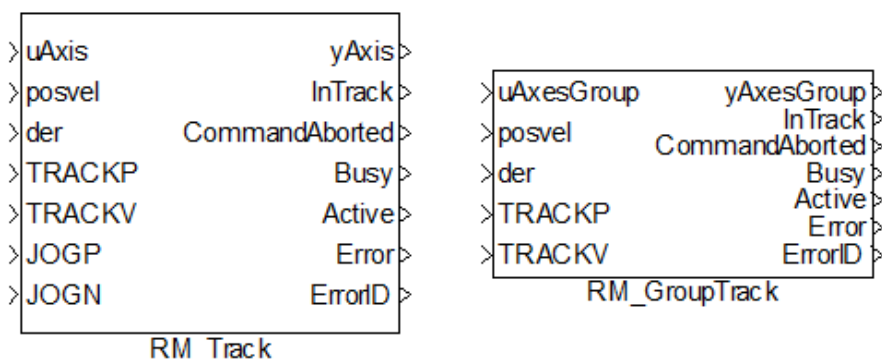
Vrstva 3: Řídicí systém pohybu

Vlastní konfigurace řídicího systému pohybu implementovaná v systému REXYGEN v podobě programování pomocí funkčních bloků, viz Obrázek 12. Subsystémy pro řízení pohybu robotů jsou charakterizovány následovně:

Bloky pro ruční posuv robotu (JOGování):

Systémy pro ruční pojezd s klouby robotu (nekoordinovaný pohyb, pohyb v ACS) či pro pojezd s koncovým efektem robotu (koordinovaný pohyb, pohyb v MCS). Principiálně funkcionalitu rozdělit na dvě skupiny:

- Standardní JOG v ACS (Axis Coordinate System) / MCS (Machine Coordinate System), viz Obrázek 4:
 - Zajištěno bloky *RM_Track* (ACS), *RM_GroupTrack* (MCS, pro implem,implementovanou kinematiku)
- Pokročilé bloky JOGování, viz Obrázek 5:
 - Pro libovolně zadanou kinematiku (parametrizace Denavit Hartebergovými parametry)
 - Možnosti hlídání maximálních povolených rychlostí kloubů (ochrana před vstupem do singularit), ochrana limitů kloubů
 - Vhodné pro rekonfiguraci robotu (např. při testování jen částí robotických systémů)
 - Možnosti integrovat optimalizaci pohybu pro redundantní architektury manipulátorů



Obrázek 4: Bloky pro standardní JOG

Vstupy

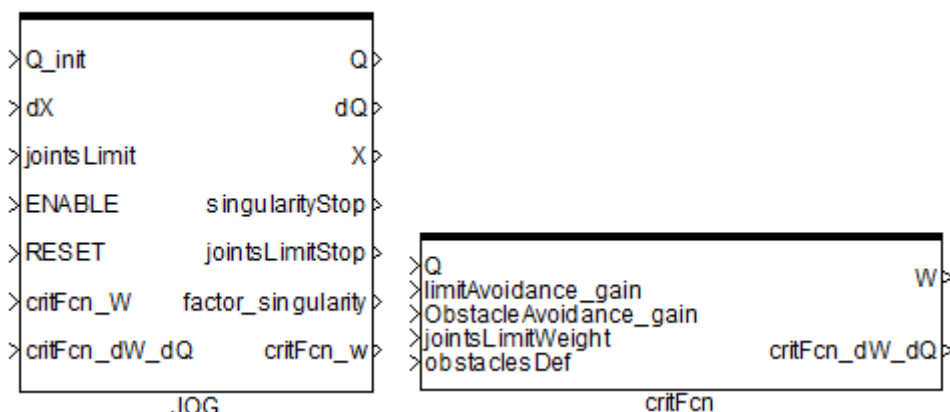
uAxis	Odkaz na osu (přípustné je jen spojení <code>RM_Axis.axisRef-uAxis</code> nebo <code>yAxis-uAxis</code>)	reference
posvel	Požadovaná poloha nebo rychlost [unit]	double
TRACKP	Zapnutí režimu sledování polohy	bool
TRACKV	Zapnutí režimu sledování rychlosti	bool
JOGP	Zapnutí režimu posunu v kladném směru	bool
JOGN	Zapnutí režimu posuvu v záporném směru	bool

Výstupy

yAxis	Odkaz na osu (přípustné je jen spojení <code>RM_Axis.axisRef-uAxis</code> nebo <code>yAxis-uAxis</code>)	reference
InTrack	Příznak dosažení požadované polohy v režimu sledování polohy	bool
CommandAborted	Příznak přerušení funkce bloku	bool
Busy	Příznak, že algoritmus ještě neskončil	bool
Active	Příznak, že blok řídí osu	bool
Error	Příznak chyby	bool
ErrorID	Kód chyby i obecná chyba systému REX	error

Parametry

pv	Maximální povolená rychlost [unit/s]	double
pa	Maximální povolené zrychlení [unit/s ²]	double
pd	Maximální povolené zpomalení [unit/s ²]	double
pj	Maximální povolená změna zrychlení [unit/s ³]	double
iLen	Počet kroků pro odhad rychlosti	⊙10 long

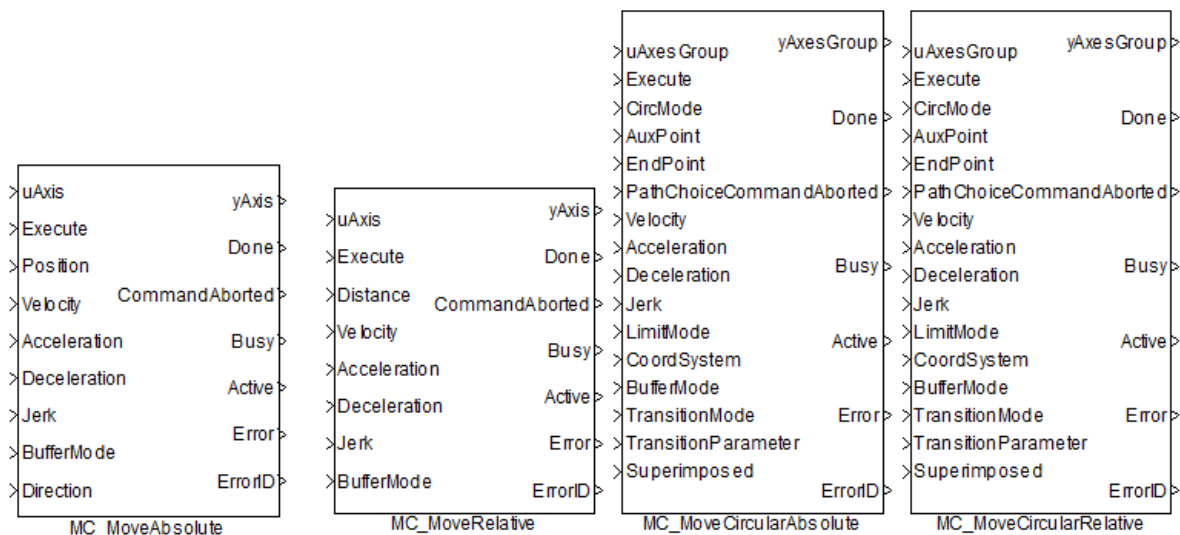


Obrázek 5: Bloky pro pokročilý JOG

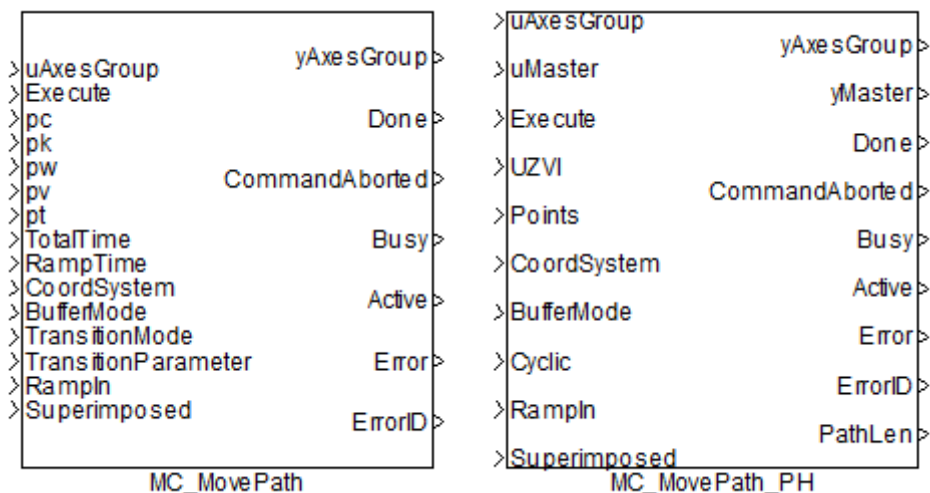
Bloky pro programový pohyb

- Standardní pohybové bloky dle PLCopen Motion Control, viz Obrázek 6:
 - Pevný program s možností absolutní/relativního lineárního/kruhového posunu do pevně zadaných bodů v ACS/MCS.
 - Implementováno bloky: `MC_MoveAbsolute`, `MC_MoveRelative`, `MC_MoveCircularAbsolute`, `MC_MoveCircularRelative`, *atd.*
- Pokročilé pohybové bloky pro realizaci hladkých prostorových trajektorií, viz Obrázek 7:
 - `MC_MovePath` - obecná po částech polynomiální/racionální interpolace s trajektorií pohybu definovanou ve tvaru NURBS funkce

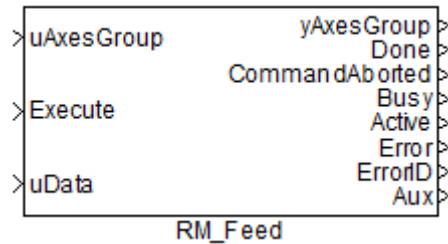
- MC_MovePath_PH - obecná po částech polynomiální funkce s Pythagorejským hodografem, speciální třída interpolačních funkcí tvořících podmnožinu obecných B-spline funkcí s možností přesného řízení definované rychlosti posuvu
- Interpretace libovolného definovaného pohybu, viz Obrázek 8
 - Realizováno blokem *RM_Feed*
 - Pohyb generován mimo jádro řídicího systému pohybu (externí aplikací, generátorem dat, atd.)
 - Rozhraní pro předávání dat: txt soubor s hodnotami polohy, rychlosti a zrychlení (ACS/MCS) v diskrétní podobě odpovídající vzorkovací periodě příslušného TASKu řídicího systému



Obrázek 6: Standardní pohybové bloky



Obrázek 9: Pokročilé pohybové bloky pro polynomiální/racionální trajektorie



Obrázek 10: Interpretace pohybu zadaného externím souborem dat

Vstupy

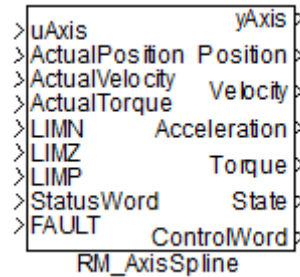
uAxis	Odkaz na osu (přípustné je jen spojení <code>RM_Axis.axisRef-uAxis</code> nebo <code>yAxis-uAxis</code>)	reference
Execute	Náběžná hrana aktivuje blok	bool
Position	Požadovaná poloha [unit]	double
Velocity	Maximální povolená rychlost [unit/s]	double
Acceleration	Maximální povolené zrychlení [unit/s ²]	double
Deceleration	Maximální povolené zpomalení [unit/s ²]	double
Jerk	Maximální povolená změna zrychlení [unit/s ³]	double
BufferMode	Režim převzetí osy	long
	1 Aborting (nový blok se spustí okamžitě)	
	2 Buffered (nový blok se spustí po dokončení předchozího)	
	3 Blending low (nový blok se spustí po dokončení předchozího, původní pohyb skončí s nižší rychlostí z obou bloků)	
	4 Blending high (nový blok se spustí po dokončení předchozího, původní pohyb skončí s vyšší rychlostí z obou bloků)	
	5 Blending previous (nový blok se spustí po dokončení předchozího, původní pohyb skončí se svojí koncovou rychlostí)	
	6 Blending next (nový blok se spustí po dokončení předchozího, původní pohyb skončí s počáteční rychlostí nového bloku)	
Direction	Směr pohybu (jen pro cyklické osy nebo speciální případy)	long
	1 kladný	
	2 nejkratší	
	3 záporný	
	4 aktuální	

Výstupy

yAxis	Odkaz na osu (přípustné je jen spojení <code>RM_Axis.axisRef-uAxis</code> nebo <code>yAxis-uAxis</code>)	reference
Done	Příznak dokončení algoritmu	bool
CommandAborted	Příznak přerušení funkce bloku	bool
Busy	Příznak, že algoritmus ještě neskončil	bool
Active	Příznak, že blok řídí osu	bool
Error	Příznak chyby	bool

Interpolátor dat

Speciálně navržený, viz Obrázek 11, blok jádra řídicího systému sloužící jako interface mezi pomalými TASKy (typicky s bloky generující pohyb robotu) a rychlým TASKem realizující komunikaci (Vrstva 2, typicky komunikace se servoměniči přes EtherCAT).



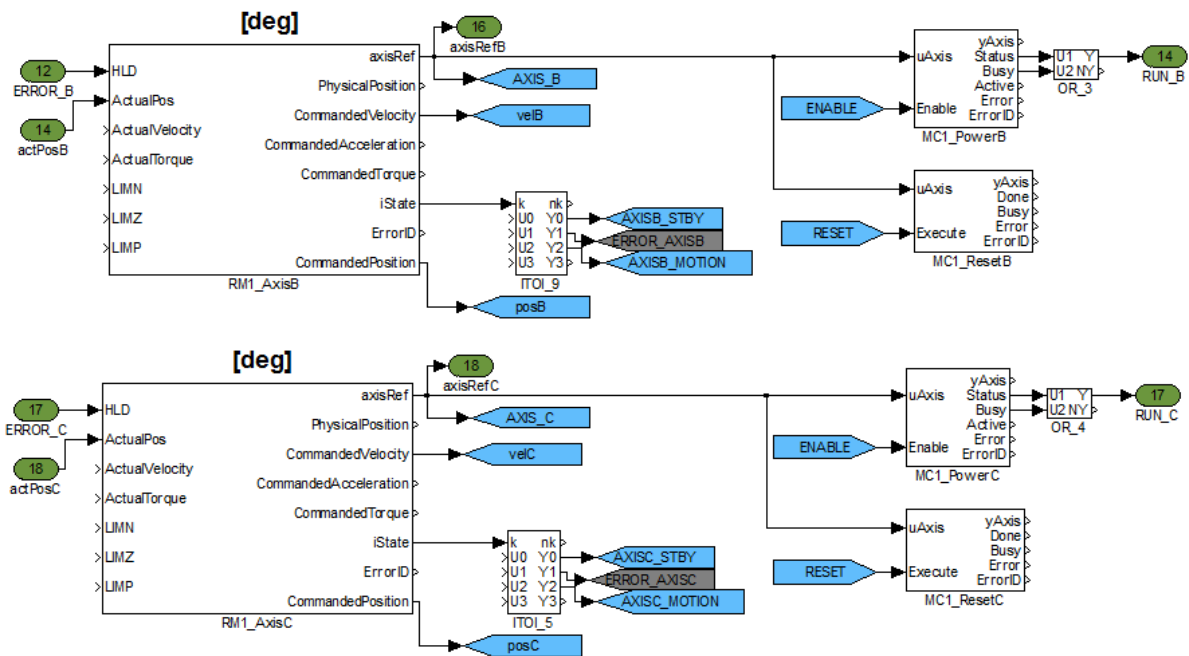
Obrázek 11: Blok interpolátoru dat

Vstup

uAxis Odkaz na osu (přípustné je jen spojení `RM_Axis.axisRef-uAxis` nebo `reference yAxis-uAxis`)

Výstupy

pos	Požadovaná (interpolovaná) poloha	double
vel	Požadovaná (interpolovaná) rychlost	double
trq	Požadovaný (interpolovaný) moment/síla	double
iState	Stav interpolátoru	long
	0 neaktivní	
	1 interval0	
	2 interval1	
	3 interval2	
	5 změna 1. predikce	
	-1 změna aktuální hodnoty	
	-2 zpoždění <code>RM_Axis</code>	
	-3 přetížení tasku	
	-4 velká perioda	



Obrázek 12: Část jádra řídicího systému v podobě funkčních bloků pro řízení os robotu

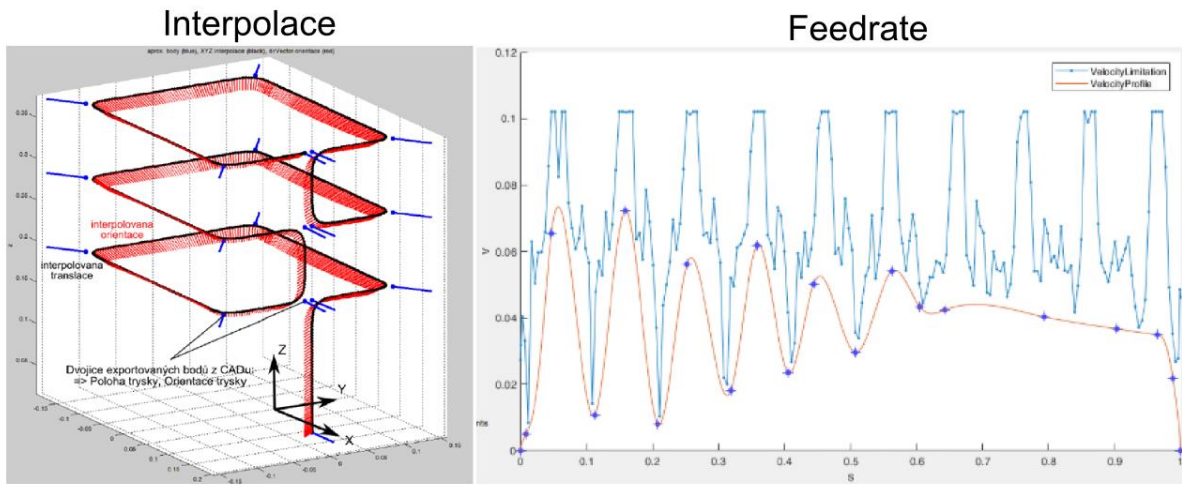
Vrstva 4: Parametrizace pohybu, Simulační ověření pohybu

Vrstva definující tvar a parametry pohybu podle zadané informace (např. definice koincidenčních bodů z nadřazené vrstvy řídicího systému, Vrstva 5) => definice kompletní informace o požadovaném pohybu robotu. Výsledná trajektorie pohybu lze odsimulovat na virtuálním simulačním modelu, který je pro každý typ robotu k dispozici:

Generátor trajektorie

Implementováno v Matlabu, viz Obrázek 13, v podobě editoru generátoru trajektorie pohybu. Subsystem složený ze dvou součástí:

- **Interpolace:** Interpolování zadaných koincidenčních bodů, řešeno jako smíšený problém interpolace/aproximace zadaných koincidenčních bodů z nadřazené vrstvy (tzn. lze parametrizovat - vyhlazovat tvar trajektorie ve smyslu zadávání váhového koeficientu plynule měnící výpočet mezi aproximací/interpolací)
 - Implementován jako funkce v Matlabu
 - Součást editoru plánovače trajektorií
 - Vstupem jsou zadané koincidenční body
 - Výstupem je (geometrická) parametrizace trajektorie pohybu
- **Feedrate:** Parametrizace pohybu podél interpolované trajektorie ve smyslu definice požadovaného profilu rychlosti (zrychlení) pohybu
 - Zohlednění maximálních rychlostí aktuátorů manipulátoru
 - Automatické tvarování profilu rychlosti
 - Ruční korekce profilu rychlosti (tvarování)

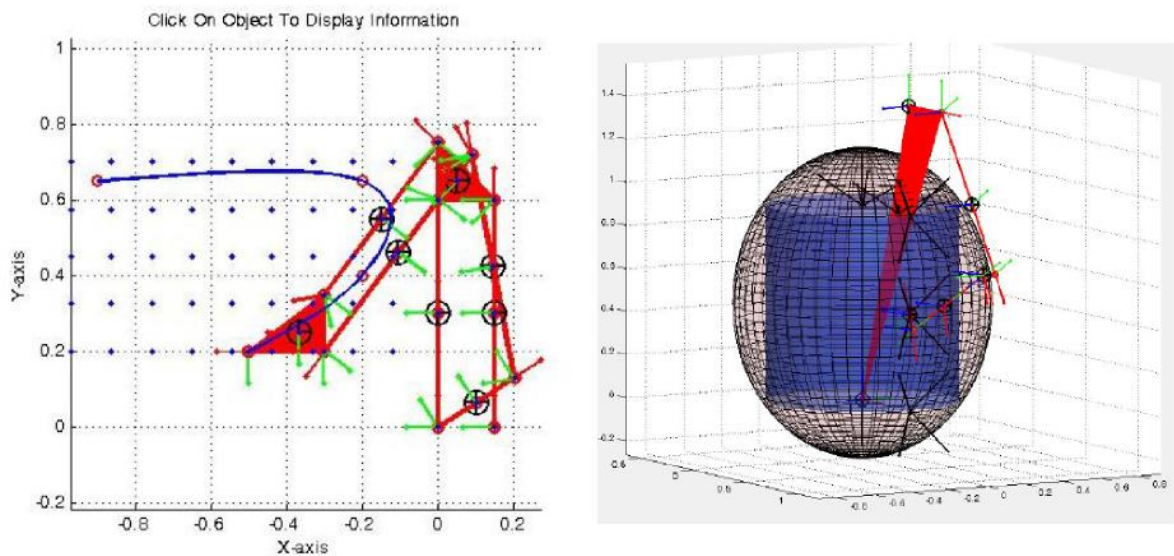


Obrázek 13: Výstupy generátoru trajektorie dat v Matlabu

Virtuální simulační model

Subsystem reprezentující virtuální simulační model v prostředí Matlab/Simulink/SimMechanics, viz Obrázek 14, a slouží k ověření plánovaných trajektorií pohybu ve smyslu:

- Vizualizace pohybu robotu v technologickém prostředí (včetně CAD modelů tech. komponent)
- Detekce kolizních stavů
- Ověření požadavků na aktuátory (rychlosti, zrychlení, momenty, atd.)



Obrázek 14: Vizualizace virtuálních simulačních modelů v SimMechanics

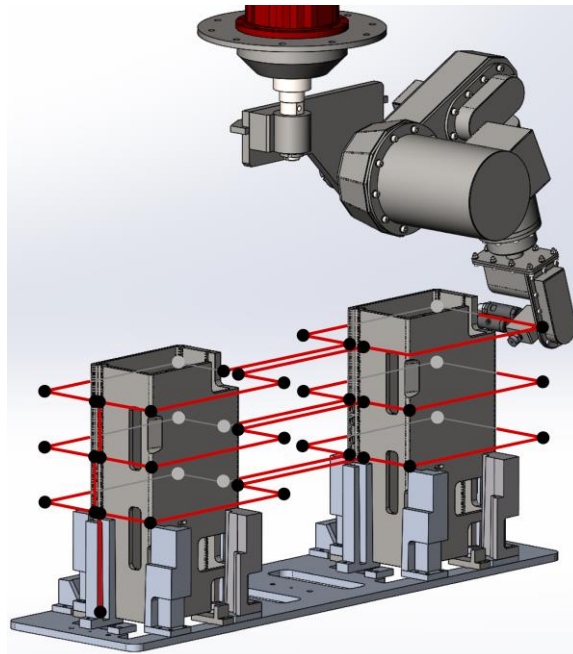
Vrstva 5: Pokročilé plánování pohybu, Konfigurace, Operátorské ovládání

Nejvyšší vrstva řídicího systému zahrnující prvotní operátorské plánování pohybu robotu, konfiguraci robotického systému a jeho uživatelské ovládání prostřednictvím operátorského panelu:

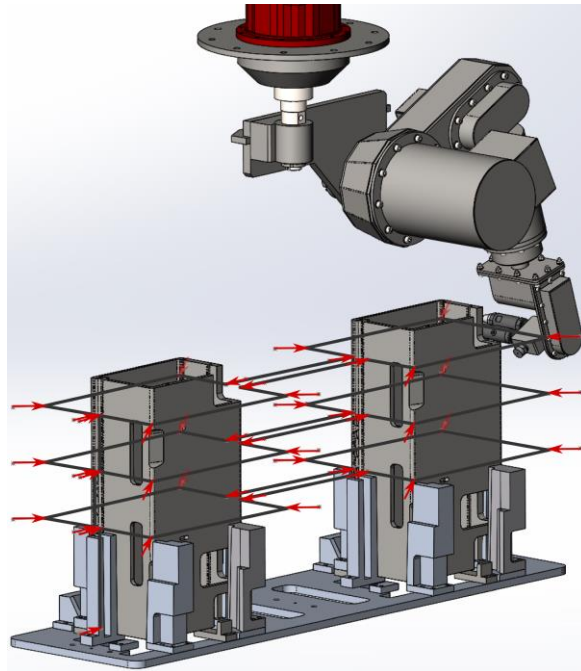
Pokročilé plánování pohybu:

Subsystem tvořen především rozhraním pro export zájmových (koincidenčních) bodů z CAD systémů, ve kterém jsou modelovány technologické díly, které mají být robotem obsluhovány. Operátor (v našem případě konstruktér těchto technologických dílů) pak může přímo v prostředí CAD systému (SolidWorks, Inventor, atd.) definovat geometrický tvar pohybové trajektorie robotu a některé další vlastnosti (např. směr trysky na koncovém efektoru robotu, atd.). Exportované zájmové body jsou pak zpracovány generátorem dat (interpolace/aproximace a parametrizace pohybu) a výsledný pohyb ověřen na virtuálním simulačním modelu (viz Vrstva 4). Technologický princip lze shrnout následovně (příklad při použití CAD systému Solidworks):

- Návrh zájmových bodů v CAD systému. U systému Solidworks je před samotným exportem zájmových bodů nutné všechny tyto body převést do jedné skici (popřípadě 3D skici, pokud body nejsou v jedné rovině) a to v pořadí, v jakém chceme, aby byla trajektorie projeta. V případě potřeby vyexportovat k trajektorii doplňující informace, např. směrové vektory určující orientaci koncového efektoru podél trajektorie, je nutné tyto vektory zanést do další skici. Pro správný převod vektorové informace je opět nutné zanést do skici vektory ve správném pořadí, přičemž vektor zakreslujeme jako dvojici bodů v pořadí začátek vektoru a konec vektoru.

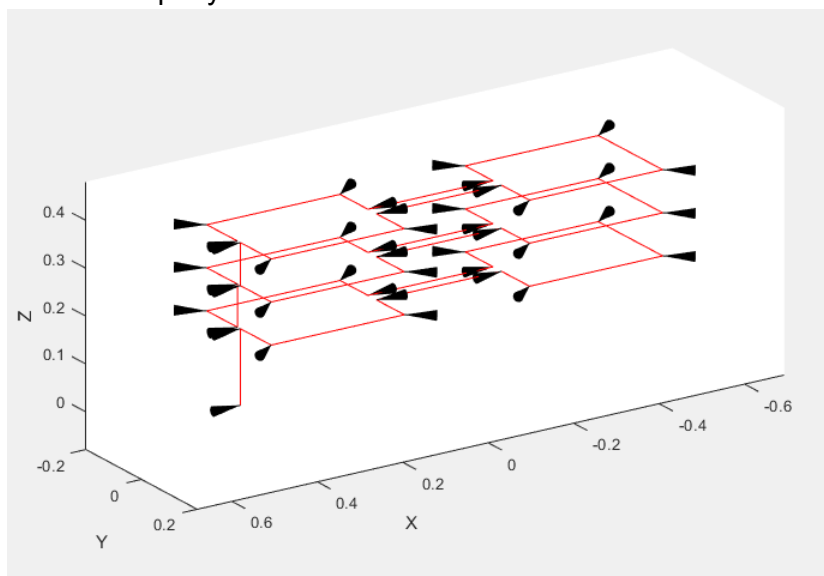


Obrázek 15: Vyznačená trajektorie procházející zájmovými body navrženými v programu Solidworks



Obrázek 16: Vyznačené navržené vektory (např. požadovaná orientace konc.efektoru) v programu Solidworks

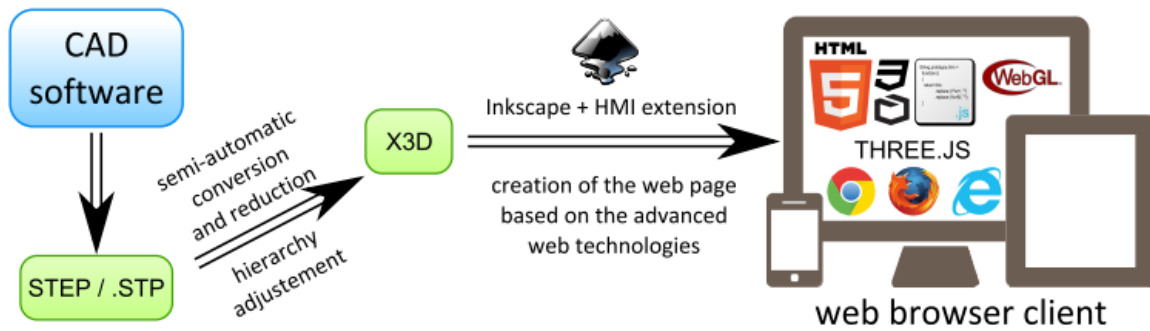
- Export zájmových bodů. V případě systému Solidworks je export bodů ze skici možný skrze makro do tabulky v programu Excel.
- Načtení exportovaných bodů v programu Matlab. Od tohoto kroku jsou již zájmové body k dispozici pro další zpracování popsané ve “Vrstva 4: Parametrizace pohybu, Simulační ověření pohybu”.



Obrázek 17: Trajektorie a vektory převedené z programu Solidworks do Matlabu

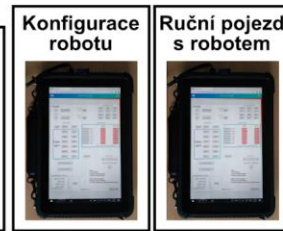
Vizualizace/operátorské pracoviště

Subsystem HMI pro interakci s obsluhou robotu implementovaný standardizovaným přístupem tvorby vizualizací v prostředí REXYGEN (včetně využití možnosti 3D vizualizace robotu v technologickém provozu - modely v CAD). Rozhraní vizualizace využívá nástroje založené na otevřených webových technologiích pro 3D grafiku vyvinuté v předchozích etapách projektu [10] .

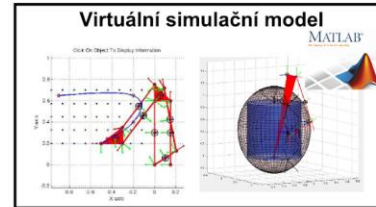
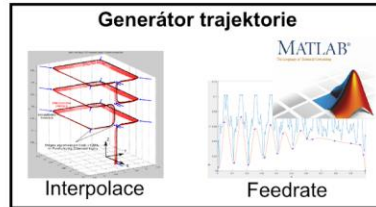


Obrázek 18: Tvorba HMI rozhraní

Vrstva 5
 Pokročilé plánování pohybu
 Konfigurace
 Operátorské ovládání



Vrstva 4
 Parametrizace pohybu
 Simulační ověření pohybu



Programový pohyb

uAxisGroup	yAxisGroup	zAxisGroup	xAxisGroup
Execute	Done	Done	Done
Position	Position	Position	Position
Velocity	Velocity	Velocity	Velocity
Acceleration	Acceleration	Acceleration	Acceleration
LimitMode	LimitMode	LimitMode	LimitMode
CoordSystem	CoordSystem	CoordSystem	CoordSystem
TransitionMode	TransitionMode	TransitionMode	TransitionMode
TransitionParameter	TransitionParameter	TransitionParameter	TransitionParameter
Superimposed	Superimposed	Superimposed	Superimposed
MC_MoveLinearAbsolute	MC_MoveLinearRelative	MC_MoveCircularAbsolute	MC_MoveCircularRelative

Standardní pohybové bloky (Lineární, kruhový pohyb, ...)

Interpretace ext. definovaného pohybu

REXYGEN

Interpolátor pohybu

uAxis	yAxis
ActualPosition	Position
ActualVelocity	Velocity
ActualTorque	Torque
LIMN	Acceleration
LIMP	Torque
StatusWord	State
FAULT	ControlWord
RM_AxisSpline	

Vrstva 3
 Řídící systém pohybu

JOGovací pohyb (ruční posuv)

Standardní JOG (ACS/MCS)

Konfigurovatelný JOG (MCS)

Vrstva 2
 Komunikace se servoměniči
 Komunikace s periferiemi

Komunikační subsystém

REXYGEN **CAN** **EtherCAT**

Vrstva 1
 Výkonová elektronika
 Servoměniče



Vrstva 0
 Mechanika robotů



Obrázek 19: Hierarchické uspořádání vrstev řídicího systému

Reference

- [1] REXCONTROLS. REXYGEN (Programming Automation Devices without Hand Coding). 2018, <https://www.rexygen.com/>.
- [2] PLCopen Motion Control Library documentation, http://www.plcopen.org/pages/tc2_motion_control/marketing_introduction/index.htm
- [3] ŠVEJDA, Martin. Kinematical, kinetostatical and dynamical analysis of 4DoF manipulator, parametric optimization of mechanical construction (DV026), 2015
- [4] ŠVEJDA, Martin. Kinematical, kinetostatical and dynamical analysis of 7DoF manipulator, parametric optimization of mechanical construction (DV027), 2015.
- [5] KŘÍŽOVÁ, Kamila, DENK, Milan, GOUBEJ, Demetr, Manipulátory pro agresivní prostředí, EuroTec JKR, 2018
- [6] KŘÍŽOVÁ, Kamila, DENK, Milan, GOUBEJ, Demetr, Manipulátory pro agresivní prostředí (D5.6), EuroTec JKR, 2017
- [7] JÁGER, Arnold, End-effector force control for special robots operating in chemically aggressive environments, ZČU v Plzni, 2017
- [8] JEŽEK, Ondřej, Ovladače průmyslových komunikací (D3.29), ZČU, 2015
- [9] JEŽEK, Ondřej, Uživatelská dokumentace SW ovladačů průmyslových komunikací s moderními řídicími jednotkami serovměničů, ZČU, 2015
- [10] SEVERA, Ondřej, GOUBEJ, Martin, KONIGSMARKOVÁ, Jana, Unified framework for generation of 3D web visualization for mechatronic systems, Journal of Physics, 2015