

# TAČR Program Gama

Název projektu:

Podpora komerčních příležitostí ZČU (TG02010011)

Dílčí projekt:

Řízení pohybu průmyslového robotu na základě procesní veličiny

Dokumentace k výsledku TG02010011-2017V002  
„Řídicí systém průmyslového robotu na základě  
procesní veličiny“

Tomáš Čechura, Martin Švejda

21. června 2017

**T A**  
**Č R**

Program **Gama**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Požadavky na pokročilé funkce řízení pohybu . . . . .	3
<b>2</b>	<b>Pokročilé funkce řízení pohybu a procesní veličiny</b>	<b>3</b>
2.1	Řízení podle procesní veličiny (síly/momentu) . . . . .	8
2.2	Postprocessing zaznamenaných trajektorií . . . . .	9
<b>3</b>	<b>Implementace řídicího systému</b>	<b>10</b>
3.1	Popis řídicího systému REX . . . . .	10
3.2	Řízení pohybu sedmiosého průmyslového robotu . . . . .	11
3.3	Tenzometrické měření - F/T senzor . . . . .	11
3.4	Uživatelské rozhraní a možnosti integrace . . . . .	11
<b>4</b>	<b>Závěr</b>	<b>12</b>

# 1 Úvod

Předkládaný dokument slouží jako dokumentace k řídicímu systému průmyslového robotu na základě procesní veličiny. Budou představeny důležité technické a uživatelské aspekty použití vzniklého řídicího systému. Při vlastním vývoji byl kladen důraz na univerzálnost a znovupoužití všech vzniklých funkčních celků. Implementací v otevřeném řídicím systému reálného času REX [1] bylo dosaženo nezávislosti vzniklých algoritmů na cílovém průmyslovém robotu a zároveň na použitém řídicím počítači (obvykle průmyslové PC). Při migraci řídicího systému na jiný průmyslový robot je nezbytné znovu vyřešit pouze nejnižší komunikační vrstvu. Všechny nadřazené vrstvy jsou obecně použitelné.

## 1.1 Požadavky na pokročilé funkce řízení pohybu

Standardní prostředky řízení pohybu jsou využívány nejen všemi výrobci průmyslových robotů. Pro vyvinutí pokročilých funkcí řízení pohybu je nutné znát nějakou další veličinu navíc oproti běžně používanému přístupu. Běžný přístup řízení pohybu zpracovává informace z enkodérů a na jejich základě určuje akční zásah.

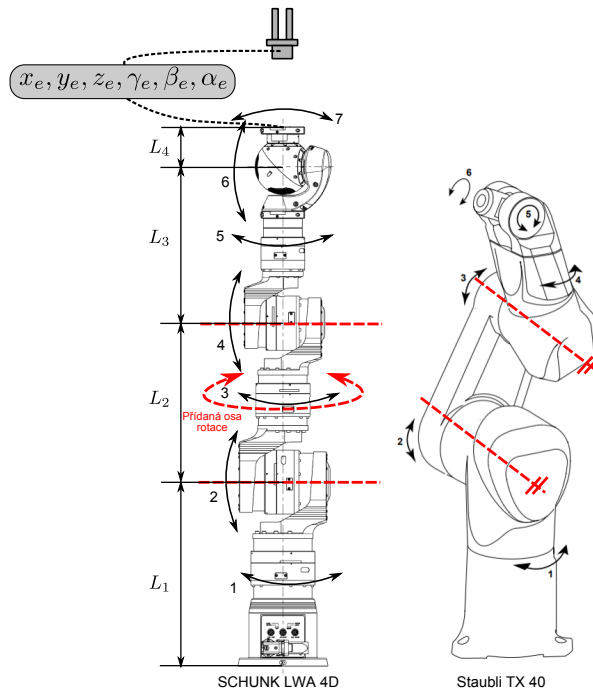
Díky neustálému vývoji vznikají stále nové typy senzorů pro použití v robotice. Mezi senzory, jejichž obliba roste, patří zejména tenzometrické buňky schopné měřit sílu a moment síly. Právě síla je jedna z hlavních procesních veličin, která poskytuje relevantní informaci o právě probíhajícím robotickém procesu. Vhodnými procesy jsou zejména kontaktní činnosti - např. indukční kalení, broušení, svařování, defektoskopické testování, apod.

## 2 Pokročilé funkce řízení pohybu a procesní veličiny

Následující kapitola shrnuje teoretické poznámky k implementovanému řídicímu SW kontrolleru pro zvolený typ robotu SCHUNK LWA 4D. Uvažovaný robot se liší od standardních 6ti-osých průmyslových robotů přidáním (redundantním) kloubem mezi 2. a 3. osou, viz Obrázek 1 (srovnání se standardním 6 DoF robotem Stäubli).

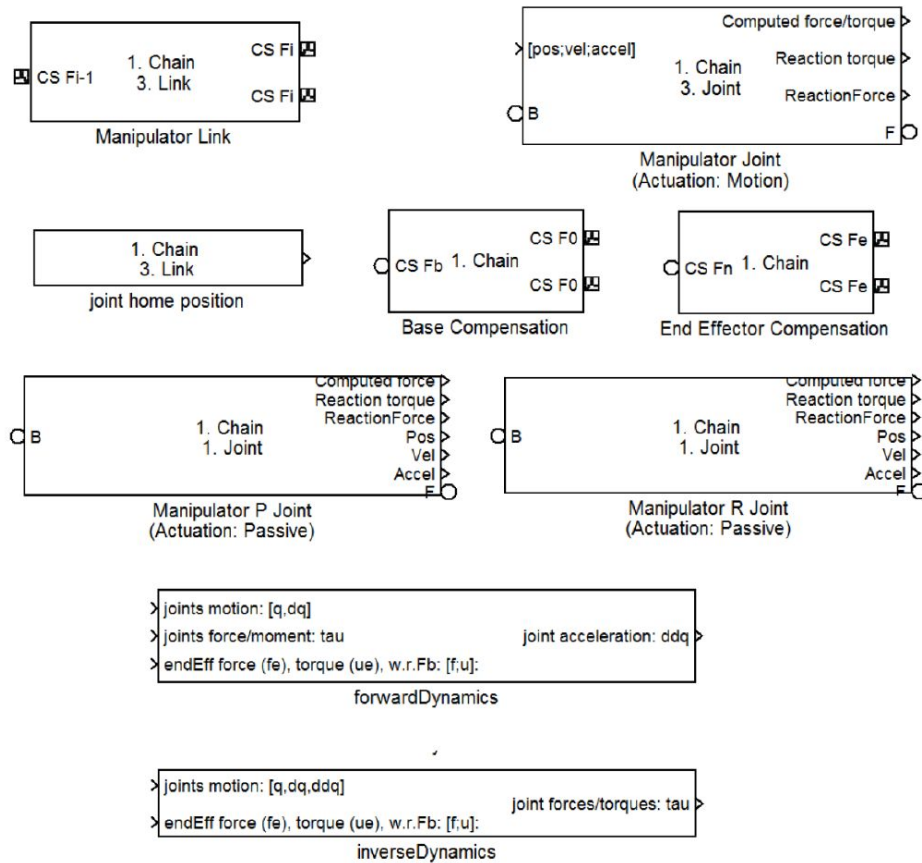
Za účelem vývoje nových pokročilých algoritmů řízení pohybu bylo nutné vytvořit virtuální simulační model manipulátoru v programové prostředí Matlab/Simulink/SimMechanics. Virtuální simulační model byl vytvořen prostřednictvím knihovny robotLib [2], která zahrnuje základní funkční bloky v prostředí SimMechanics pro tvorbu sériových a rozvětvených kinematických řetězců, viz Obrázek 2. Dále jsou v knihovně zahrnuty některé předimplementované elementární funkce pro podporu kinematických a dynamických úloh manipulátorů (dopředná/zpětná kinematika, přímý/inverzní dynamický model) a některé další podpůrné funkce, např.: převody mezi reprezentacemi orientace, funkce pro parametrizaci sestavených modelů manipulátorů pomocí předimplementovaných funkcí v SimMechanics, elementární funkce pro plánování trajektorií pohybu, atd.

Uvažovaný manipulátor SCHUNK LWA 4D lze z hlediska kinematiky plně popsat tzv. Denavit-Hartbergovou úmluvou (metodologie přiřazení souřadných systémů (s.s.) ramenům manipulátoru) a výsledný kinematický model manipulátoru poté parametrizovat pouze délkami jednotlivých ramen  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$  a případnými kompenzacemi polohy koncového efektoru (statická transformace reprezentující umístění nástroje na koncovém



Obrázek 1: Porovnání architektury 7 DoF robota SCHUNK LWA 4D se standardním 6 DoF průmyslovým robotem

efektoru manipulátoru - standardně 6 parametrů:  $x_e, y_e, z_e$  pro translaci a  $\gamma_e, \beta_e, \alpha_e$  pro rotaci v podobě ZYX Eulerových úhlů).



Obrázek 2: Standardní funkční bloky knihovny robotLib

Kloubové souřadnice manipulátoru jsou reprezentovány vektorem

$$\mathbf{Q} = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6 \ q_7]^T \quad (1)$$

Zobecněné souřadnice (poloha koncového efektoru) manipulátoru ve smyslu translace a rotace reprezentované ZYX Eulerovými úhly jsou dány jako

$$\mathbf{X} = [x \ y \ z \ \gamma \ \beta \ \alpha]^T \quad (2)$$

Klíčové kinematické úlohy manipulátoru byly realizovány následujícími funkcemi:

`[genCoords] = forwardKinematics(jointCoords, kinPar, N):`

Výpočet přímé kinematické úlohy (polohy) a okamžité přímé kinematické úlohy (rychlosti, zrychlení). Obecně platná pro sériové manipulátory a lze počítat algoritmicky:

$$\{\mathbf{Q}, \dot{\mathbf{Q}}, \ddot{\mathbf{Q}}\} \Rightarrow \{\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}\} \quad (3)$$

<u>Vstupy:</u>	
<code>jointCoords</code>	Hodnoty poloh kloubových souřadnic, odpovídajících rychlostí a zrychlení, ve formátu: $[\mathbf{Q}, \dot{\mathbf{Q}}, \ddot{\mathbf{Q}}]$
<code>kinPar</code>	Kinematické parametry manipulátoru, viz funkce <code>robotSetup</code> (viz knihovna <code>robotLib</code> a inicializace kinematické architektury robotu)
<code>N</code>	Pro $N = j + 1$ bude výstupem funkce poloha, rychlost a zrychlení $j$ -tého ramena, tedy s.s. $F_j$ , $j = 1, 2, \dots, n$ , pro $N = 1$ resp. $N = n + 2$ bude výstupem poloha, rychlost, zrychlení 0-tého s.s. $F_0$ resp. s.s. konc. efektoru $F_e$ . Pokud $N$ nezadáno, vracena poloha, rychlost a zrychlení s.s. konc. efektoru $F_e$

<u>Výstupy:</u>	
<code>genCoords</code>	Polohy, rychlosti a zrychlení zobecněných souřadnic (závisí na vstupu <code>N</code> ) odpovídajícího s.s. vzhledem k s.s. $F_b$ , např. pro $N$ nezadáno je výstup ve formátu $[[\mathbf{O}_e^b, \mathbf{R}_e^b], [\dot{\mathbf{O}}_e^b, \boldsymbol{\omega}_e^b], [\ddot{\mathbf{O}}_e^b, \dot{\boldsymbol{\omega}}_e^b]]$ (poloha, rychlost, zrychlení s.s. efektoru v s.s. základny manipulátoru)

`[Q,w,condJ] = inverseKinematics(Xd,Q_init,kinPar, jointCoords_limit):`

— nově implementovaná funkce nad rámec knihovny `robotLib` —

Výpočet inverzní kinematické úlohy (polohy):

$$\mathbf{X} \Rightarrow \mathbf{Q} \quad (4)$$

Vzhledem ke kinematické redundanci robotu není výpočet inverzní kinematické úlohy jednoznačný (existuje nekonečně mnoho řešení). Právě proto lze do výpočtu zahrnout sekundární kritérium optimality, které definuje míru optimality aktuální konfigurace robotu. Interní iterační algoritmus ve funkci poté konverguje k takovému řešení, které lokálně maximalizuje zvolené kritérium optimality.

Vzhledem k podstatě úlohy bylo voleno kritérium optimality, které zohledňuje záporně vzatou hodnotu součtu kvadrátů vzdálenosti dílčích kloubů manipulátoru od středu povoleného rozsahu pohybu:

$$\begin{aligned}
 w(\mathbf{Q}) &= -k_0 \cdot \frac{1}{2n} \sum_{i=1}^n \left( \frac{q_i - \bar{q}_i}{q_i^{max} - q_i^{min}} \right)^2, \quad n = 7 \\
 \mathbf{Q} &= [q_1, q_2, \dots, q_n]^T \\
 \mathbf{Q}^{min} &= [q_1^{min}, q_2^{min}, \dots, q_n^{min}]^T \\
 \mathbf{Q}^{max} &= [q_1^{max}, q_2^{max}, \dots, q_n^{max}]^T
 \end{aligned} \quad (5)$$

kde  $\bar{q}_i = q_i^{min} + \frac{q_i^{max} - q_i^{min}}{2}$  a  $q_i^{max}$  resp.  $q_i^{min}$  označuje maximální resp. minimální polohu kloubu.

Maximalizací kritériální funkce tedy algoritmus výpočtu vnitřně rekonfigurujeme mechaniku manipulátoru takovým způsobem, aby se klouby manipulátoru udržovaly co nejdále od svých limitních poloh. Volbou konstanty  $k_0$  vážíme kompromis mezi udržováním kloubů ve středech svých povolených rozsahů ( $k_0 \gg 0$ ) a minimalizací kloubových rychlostí ( $k_0 \rightarrow 0$ ) - nativní vlastnost interního iteračního algoritmu pro  $w(\mathbf{Q}) = \mathbf{0}$ .

<u>Vstupy:</u>	
$\mathbf{X}_d$	Požadovaná poloha koncového efektoru
$\mathbf{Q}_{init}$	Počáteční polohy kloubových souřadnic (počáteční podmínka numerického algoritmu)
$\mathbf{kinPar}$	Kinematické parametry manipulátoru, viz výše
$\mathbf{jointCoords\_limit}$	Limity kloubových souřadnic $\langle \mathbf{Q}_{min} \mathbf{Q}_{max} \rangle$
<u>Výstupy:</u>	
$\mathbf{Q}$	Vypočtené polohy kloubových souřadnic
$w$	Hodnota kritériální funkce pro nalezené řešení
$\mathbf{condJ}$	Převrácená hodnota čísla podmíněnosti kinematického jakobiánu manipulátoru v poloze $\mathbf{Q}$

`[dQ] = JOG_limitsOvercoming(Q,Xd,dXd,kinPar,Q_lim):`

— nově implementovaná funkce nad rámec knihovny robotLib —

Výpočet inverzní kinematické úlohy pro kloubové rychlosti ze zadané polohy a rychlosti koncového efektoru. Algoritmus výpočtu využívá vstupní data o poloze a rychlosti koncového efektoru vzhledem k aktuálnímu s.s. koncového efektoru a generuje optimální požadované rychlosti kloubů manipulátoru (shodné kritérium optimalizace jako v předchozím případě). Algoritmus lze snadno modifikovat takovým způsobem, aby nebyly potřebné polohové vstupy  $\mathbf{X}_d$ , ale pouze rychlostní vstupy  $\dot{\mathbf{X}}_d$  (zjednodušená podoba algoritmu). V takovém případě sice nelze zaručit konvergenci k požadované poloze  $\mathbf{X}_d$ , nicméně v případě využití algoritmu pouze k funkci optimálního JOGu (koordinovaný přesun koncového efektoru dle zadaných požadavků operátora např. pomocí teachpendantu) lze takovou limitaci akceptovat (polohová zpětná vazba je v podstatě realizována samotným operátorem).

$$\{\mathbf{X}_d, \dot{\mathbf{X}}_d\} \Rightarrow \dot{\mathbf{Q}} \quad (6)$$

<u>Vstupy:</u>	
$\mathbf{Q}$	Aktuální poloha robotu daná kloubovými polohami (stav)
$\mathbf{X}_d$	Požadovaná poloha koncového efektoru (volitelně, nemusí být zadána)
$d\mathbf{X}_d$	Požadovaná rychlost koncového efektoru
$\mathbf{kinPar}$	Kinematické parametry manipulátoru, viz výše
$\mathbf{Q\_lim}$	Limity kloubových souřadnic $\langle \mathbf{Q}_{min} \mathbf{Q}_{max} \rangle$
<u>Výstupy:</u>	
$d\mathbf{Q}$	Optimální kloubové rychlosti

## 2.1 Řízení podle procesní veličiny (síly/momentu)

Procesní veličinou je v tomto případě 6 DoF silové/momentové měření na F/T senzoru osazené mezi posledním ramenem manipulátoru a koncovým efektozem (uchopovačem). Nedílnou součástí možnosti odměřování externích silových/momentových interakcí koncového efektoru s překážkou či operátorem je odhad hmotnostních parametrů všech komponent za měřicí částí F/T senzoru (konkrétně např. pohyblivé části F/T senzoru, uchopovače, konstrukčních prvků či samotného neseného břemene). Za tímto účelem byly implementovány následující funkce:

```
[M,ree] = endEffMassCG_estim(Fee_vect,Mee_vect,EA_zyx_vect):
```

— nově implementovaná funkce nad rámec knihovny robotLib —

Funkce pro výpočet odhadu hmotnostních parametrů (hmotnost a umístění těžiště) pro kompenzaci statických sil/momentů působících na F/T senzor v důsledku rozmístění hmotnosti za měřicí částí F/T senzoru. Neodhaduje se moment setrvačnosti, neboť lze předpokládat, že v uvažovaných aplikacích „hand guidance“ se bude koncový efektor robotu pohybovat pouze velmi pomalu.

Vstupy:

Fee_vect	Vektor měřených sil F/T senzorem v s.s. F/T senzoru [[ $F_x^1, F_y^1, F_z^1$ ] <sup>T</sup> , [ $F_x^2, F_y^2, F_z^2$ ] <sup>T</sup> , ...]
Mee_vect	Vektor měřených silových momentů F/T senzorem v s.s. F/T senzoru [[ $M_x^1, M_y^1, M_z^1$ ] <sup>T</sup> , [ $M_x^2, M_y^2, M_z^2$ ] <sup>T</sup> , ...]
EA_zyx_vect	Vektor odpovídajících orientací koncového efektoru zadaný jako ZYX Eulerovy úhly v s.s. základny [[ $\gamma^1, \beta^1, \alpha^1$ ] <sup>T</sup> , [ $\gamma^2, \beta^2, \alpha^2$ ] <sup>T</sup> , ...] odpovídající naměřeným silovým/momentovým datům

Výstupy:

M	Identifikovaná hmotnost za F/T senzorem
ree	Identifikované umístění těžiště v s.s. F/T senzoru

```
[Fee,Mee] = endEffFT_recomp(EA_zyx,M,ree):
```

— nově implementovaná funkce nad rámec knihovny robotLib —

Funkce pro výpočet kompenzace síly/silového momentu působících na F/T senzor v důsledku rozmístění hmotnosti za měřicí částí F/T senzoru. Hodnoty síly a silového momentu jsou potom použity pro odečtení ze skutečně naměřených hodnot síly a silového momentu měřeného F/T senzorem. Jako výsledek dostáváme skutečné síly a silové momenty vznikající na F/T senzoru pouze v důsledku vnějších působících sil/silových momentů.



Vstupy:

- EA\_zyx** Vektor aktuální orientace koncového efektoru zadaný jako ZYX Eulerovy úhly v s.s. základny  $[\gamma, \beta, \alpha]^T$
- M** Hmotnost za F/T senzorem (identifikované, viz výše)
- re** Umístění těžiště v s.s. F/T senzoru (identifikované, viz výše)

Výstupy:

- Fee** Vypočtený vektor síly  $[F_x, F_y, F_z]^T$  působící na F/T senzor v zadané orientaci koncového efektoru v důsledku rozmístění hmotnosti za měřící částí F/T senzoru (hodnota pro kompenzaci skutečně měřených dat F/T senzorem)
- Mee** Vypočtený vektor silového momentu  $[M_x, M_y, M_z]^T$  působící na F/T senzor v zadané orientaci koncového efektoru v důsledku rozmístění hmotnosti za měřící částí F/T senzoru (hodnota pro kompenzaci skutečně měřených dat F/T senzorem)

Řízení dle procesní veličiny (síly/silového momentu) bylo dále implementováno do řídicího systému kontroléru manipulátoru ve dvou variantách (obě využívající provedené kompenzace, viz výše):

- **Hand guidance:** Tedy navádění koncového efektoru manipulátoru rukou operátora tažením za koncový efektor umístěný za F/T senzorem, kdy naměřená kompenzovaná data síly a silového momentu byla přímo využita pro generování požadovaného pohybu koncového efektoru.
- **Bezpečnostní subsystém robotu:** Naměřená data z F/T senzoru jsou využita k detekci kolizí koncového efektoru s prostředím/operátorem.

## 2.2 Postprocessing zaznamenaných trajektorií

Optimální JOG realizovaný funkcí `JOG_limitsOvercoming` byl využit pro učení zájmových bodů pro plánování trajektorie robotu. Po naučení bylo nutné ze získaných zájmových bodů vygenerovat požadované pohyby koncového efektoru manipulátoru s následujícími požadavky:

- Mezi zájmovými body se bude přejíždět po přímkách s definovaným profilem rychlosti
- Koordinovaný pohyb bude stále optimálním pohybem dle výše uvedeného kritéria optimality (tzn. budou využity implementované funkce pro výpočet inverzní kinematické úlohy)
- Musí být zajištěn požadavek, že trajektorie skončí přesně v bodě odpovídající naučeným kloubovým souřadnicím, tzn. nepřipouští se jiné řešení kinematické úlohy na konci pohybu (tento požadavek vznikl v důsledku nepřesností v kalibraci kinematiky manipulátoru, kdy nelze zaručit že skutečná poloha koncového efektoru pro různá řešení - sadu kloubových souřadnic - bude zcela identická)

Výsledný algoritmus plánovače byl realizován jako skript v Matlabu a je součástí přílohy SW kontroléru manipulátoru.

### 3 Implementace řídicího systému

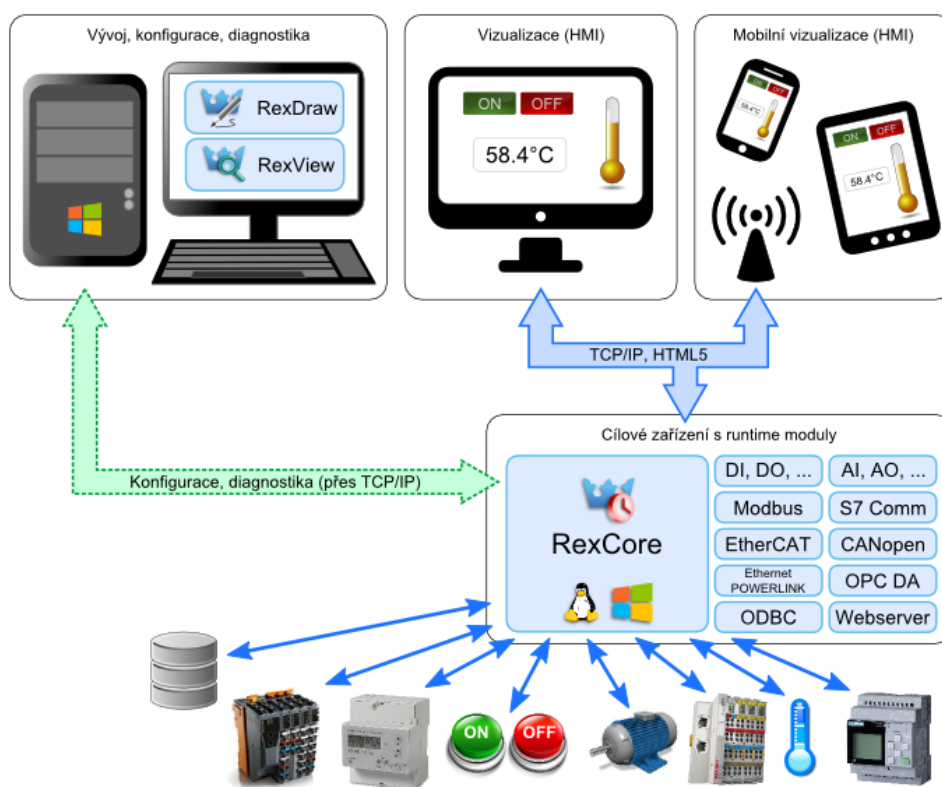
Vyvinuté algoritmy v Kapitole 2, které byly otestovány na virtuálním simulačním modelu, byly použity pro implementaci funkcí reálného řídicího systému kontroléru robotu za použití vývojového prostředí průmyslového řídicího systému REX.

#### 3.1 Popis řídicího systému REX

Řídicí systém REX slouží jako univerzální nástroj pro implementaci řídicích systémů a jejich integraci do průmyslového prostředí. Kvůli integraci disponuje podporou většiny standardně používaných komunikačních protokolů. Konkrétní projekt řídicího systému je nezávislý na cílové hardwarové platformě a je možné všechny vyvinuté algoritmy řízení jednoduše přenášet na jiné platformy.

Součástí systému REX je nativní podpora řízení pohybu mechatronických systémů. Díky integraci platné normy PLCopen Motion Control je možné řídit pohyb víceosých zařízení mezi která patří i standardní průmyslové šestiosé roboty.

Díky těmto klíčovým vlastnostem byl REX vybrán pro implementaci řídicího systému průmyslového robotu na základě procesní veličiny.



Obrázek 3: Přehled klíčových součástí řídicího systému REX

## 3.2 Řízení pohybu sedmiosého průmyslového robotu

Vzhledem k uvažovanému obecnějšímu použití jsme se rozhodli řídicí systém vyvinout primárně pro redundantní sedmiosý průmyslový robot a vše vyzkoušet na reálném sedmiosém robotu SCHUNK LWA 4D. Tato volba umožňuje použít většinu vyvinutých algoritmů jak na sedmiosých robotech, tak i na běžnějších šestiosých průmyslových robotech. Ty algoritmy, které není možné použít na standardních šestiosých robotech jsou pokročilé optimální algoritmy, které využívají právě zmíněnou redundanci prvního řádu.

Řídicí systém umožňuje zároveň pohyb robotu v jednotlivých osách i koordinovaný s využitím všech os najednou. Jsou využívány knihovny funkčních bloků pro pohyb a správu samostatné osy a zároveň pohyb více os. Nekoordinovaný pohyb se používá převážně pro operátorské navádění robotu. Koordinované řízení je mnohem intuitivnější a zároveň se používá ve většině trajektorií v automatickém režimu pohybu robotu.

Jednotlivé osy robotu jsou definovány v řídicím systému včetně důležitých parametrů odrážejících mechanické vlastnosti robotu. Zejména se jedná o nastavení typu kloubů, funkčních rozsahů všech kloubů a omezení maximálních rychlostí, zrychlení a momentů. Tato omezení jsou následně respektována v rámci celého řídicího systému včetně všech pokročilých algoritmů řízení implementovaných v rámci tohoto systému. Každá osa je opatřena funkčními bloky pro zapnutí pohybu (*MC\_Power*) a reset případných chyb na ose (*MC\_Reset*). Dále je možné zařadit různé diagnostické funkční bloky, které sledují aktuální stav osy.

Pro implementaci koordinovaného pohybu bylo nutné odvodit vztahy pro přímou a inverzní kinematickou transformaci sedmiosého sériového robotu. Tyto transformace byly integrovány do funkčního bloku *MCP\_SetKinTransform\_Schunk*. Po zapnutí řídicího systému jsou jednotlivé osy postupně přidány do grupy os, které jsou následně přiřazeny výše zmíněné kinematické transformace.

## 3.3 Tenzometrické měření - F/T senzor

Pro úspěšné vyzkoušení vyvinutých algoritmů řízení bylo nutné propojit řídicí systém s dodatečným měřením procesní veličiny. Před koncový uchopovač na efektoru robotu byla umístěna tenzometrická buňka SCHUNK FTM. Tato buňka měří aktuální působící síly v osách X, Y, Z a působící momenty sil kolem os X, Y, Z.

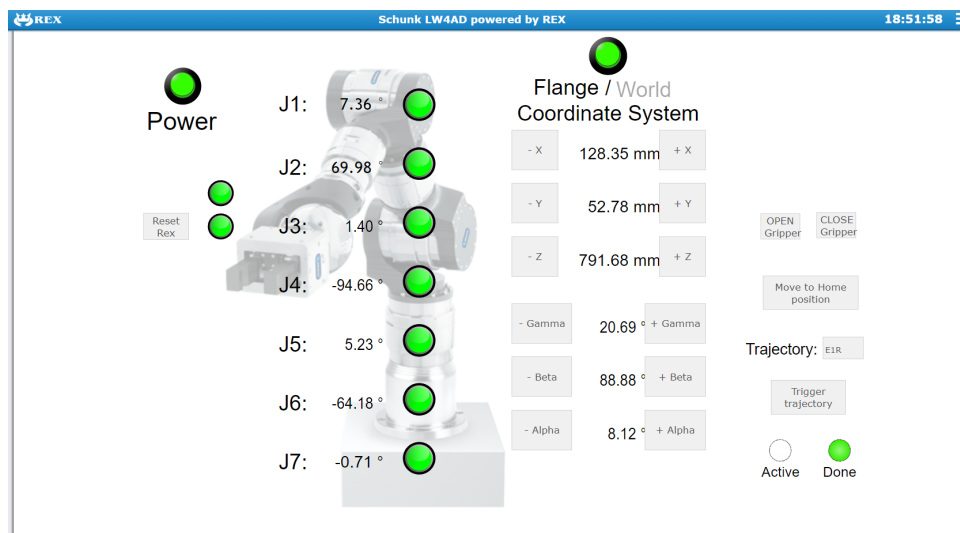
Tenzometrická buňka SCHUNK FTM disponuje rozhraním CAN. Po této sběrnici jsou přenášeny aktuální údaje z jednotlivých tenzometrů do řídicího systému s periodou 10 ms. Hodnoty ze šesti tenzometrů jsou přepočítány pomocí kalibrační matice na síly  $F_x$ ,  $F_y$ ,  $F_z$  a momenty  $T_x$ ,  $T_y$ ,  $T_z$  působící uprostřed F/T senzoru. Pro použitelnost takto získaných informací je nezbytné odhadnout hmotnostní parametry uchopovače včetně případného břemene). Následuje výpočet kompenzace síly/momentu sil v důsledku rozmístění hmotnosti za měřicí částí F/T senzoru.

## 3.4 Uživatelské rozhraní a možnosti integrace

Nedílnou součástí řídicího systému je uživatelské rozhraní - možnost pohodlně a efektivně ovládat a nastavovat nejdůležitější součásti řídicího systému přímo za běhu řídicích algoritmů. K jeho vývoji bylo použito nástroje *ReXHMI Designer*, který je standardní součástí balíku nástrojů řídicího systému REX.

*RexHMI Designer* je nadstavba profesionálního vektorového editoru Inkscape [3]. Výstupem editoru je grafický soubor s příponou .svg, který se při překladu celého projektu řídicího systému přegeneruje do podoby webové stránky využívající technologii HTML5. Takto získanou vizualizaci je možné pohodlně zobrazit v běžném webovém prohlížeči včetně prohlížečů na dnes běžných mobilních zařízeních.

Implementované uživatelské rozhraní, zobrazené na Obrázku 4, umožňuje uživateli/operátorovi přistupovat k základním možnostem řídicího systému. Vizualizace řídicího systému slouží také jako diagnostický kanál, kde zobrazované údaje podávají nepřetržitě aktuální informace o stavu robotu.



Obrázek 4: Uživatelské rozhraní řídicího systému

Operátor může aktivovat/deaktivovat pohyb robotu tlačítkem *Power*. Dále je běžnému uživateli povoleno pohybovat s robotem koordinovaným pohybem vzhledem k souřadnému systému základny robotu nebo k souřadnému systému koncového efektoru. Základní operátorské rozhraní také umožňuje spustit předdefinovanou trajektorii pohybu.

Mezi diagnostické informace patří zejména kloubové a zobecněné souřadnice robotu. Další velmi důležitou informaci podávají indikátory vzniklých chyb na jednotlivých osách.

Pokročilé funkce řídicího systému jsou přístupné přes REST API řídicího systému REX [4]. Přes REST API je možné přistupovat k ovládacím prvkům všech funkčních celků vyvinutého řídicího systému. Zároveň je možné číst důležitá data a usuzovat tak o aktuálním stavu robotu a průběhu daného algoritmu.

## 4 Závěr

Předložená zpráva dokumentuje vývoj a vznik řídicího systému průmyslového robotu s využitím znalosti procesní veličiny. Vyvinuté algoritmy se zaměřují na použití síly/silového momentu působících na koncový efektor. Byl implementován kompletní simulační model sedmiosého robotu, na kterém byly všechny pokročilé algoritmy řízení nejprve vyzkoušeny a otestovány. Po úspěšné simulační fázi jsme vytvořili řídicí systém kontroléru robotu SCHUNK LWA 4D a provedli komplexní testy na reálném průmyslovém robotu v laboratorních podmínkách.

Mezi implementované algoritmy pokročilého řízení pohybu patří *Hand guidance* - navádění koncového efektoru tažením rukou operátora, *Bezpečnostní subsystém* - online detekce kolizí koncového efektoru, *Optimální koordinovaný pohyb* - JOG s využitím redundance a s vhodným kritériem optimality, *Plánovač trajektorií zaznamenanými body* - samostatný generátor trajektorií ze získaných zájmových bodů.

Všechny vyvinuté algoritmy a funkční celky jsou univerzální a s minimálními úpravami snadno přenositelné na jiné typy průmyslových robotů s obdobnou instrumentací.

## Reference

- [1] REX Controls s.r.o., “Řídicí systém REX,” <https://www.rexcontrols.cz/rex>.
- [2] M. Švejda, “Optimalizace robotických architektur,” Disertační práce, Západočeská univerzita v Plzni 2016. [http://home.zcu.cz/~msvejda/\\_publications/2016/4\\_SvejdaMartin\\_thesis\\_2016\\_06\\_14.pdf](http://home.zcu.cz/~msvejda/_publications/2016/4_SvejdaMartin_thesis_2016_06_14.pdf).
- [3] “Inkscape,” <https://inkscape.org/en/>.
- [4] REX Controls s.r.o., “REST API of the REX Control System,” [https://www.rexcontrols.cz/media/doc/PDF/ENGLISH/RexRestApi\\_ENG.pdf](https://www.rexcontrols.cz/media/doc/PDF/ENGLISH/RexRestApi_ENG.pdf).