

MPO TRIO

Název projektu:

Výzkum a vývoj pokročilé kolaborativní robotické platformy a její aplikace ve výrobě elektronických komponent
(FV10044)

Výzkumná zpráva

Návrh, modelování a optimalizace a řízení pohybu robotického ramena (SCHUNK)

Martin Švejda (ZČU)

4. ledna 2018



MINISTERSTVO
PRŮMYSLU A OBCHODU

Identifikátor účastníka:

ZČU: Západočeská univerzita v Plzni, IMI: Integrated Micro-Electronics Czech Republic s.r.o.,
SM: SmartMotion s.r.o.

Abstrakt

Předložená technická zpráva se zabývá výzkumem pokročilého řídicího systému redundantního 7 DoF manipulátoru SCHUNK LWA 4D. Vedle realizace virtuálního simulačního modelu manipulátoru a popisu přímé a inverzní kinematické úlohy je hlavním náplní výzkum, vývoj, implementace a otestování algoritmů optimálního řízení pohybu manipulátoru s ohledem na limity jeho aktuátorů a redukci maximálních povolených kloubových rychlostí. Klíčovou výhodou redundantního manipulátoru ve spojení s navrženým řízením je zcela zásadní navýšení „obratnosti“ robotu a z toho vyplývající rozšíření pracovního prostoru, které není možné realizovat standardními architekturami 6 DoF průmyslových robotů (obdobných velikostí - zástavbového prostoru).

Obsah

1 Úvod	4
2 Virtuální simulační model manipulátoru	5
2.1 Přímá kinematická úloha	7
2.2 Inverzní kinematická úloha	7
2.3 Dynamika manipulátoru	9
3 Optimální řízení pohybu robotu	9
3.1 Učení počátečních a koncových bodů dílčích trajektorií	10
3.2 Plánování přímkového pohybu mezi naučenými body trajektorie	12
4 Závěr	17
A Parametry manipulátoru SCHUNK LWA 4D	19
A.1 7 DoF robot SCHUNK LWA 4D	19
A.2 Force/Torque jednotka FTM 45	20
A.3 Uchopovač PG+ 70	21
B Inverzní kinematika (polohové závislosti - Matlab)	22

1 Úvod

Předložená technická zpráva se zabývá vývojem řídicího systému manipulátoru firmy SCHUNK, konkrétně nabízeného typu LWA 4D. Jedná se o redundantní 7-osý manipulátor. Základní parametry nabízeného manipulátoru jsou uvedeny v příloze A.1. Manipulátor je dovybaven navíc jednotkou pro měření síly a silového momentu mezi posledním ramenem manipulátoru a uchopovačem, jedná se o model FTM 45, v příloze A.2 je pro názornost uveden dostupný datasheet jednotky FTM 75 pro větší měřicí rozsah¹. Koncovým efektoem manipulátoru je uchopovač PG+ 70, viz A.3.

Vzhledem k redundanci manipulátoru je manipulátor SCHUNK LWA 4D nestandardním manipulátorem, který se od běžných průmyslových manipulátorů liší v počtu aktivně řízených kloubů. Na Obrázku 1 je znázorněno umístění os rotace pro standardní průmyslový manipulátor (Stäubli TX 40) a uvažovaný redundantní manipulátor (SCHUNK LWA 4D). Zásadním rozdílem v kinematické architektuře obou robotů je, že redundantní manipulátor obsahuje navíc jeden rotační aktuátor, který umožňuje natáčet jinak paralelní osy rotace aktuátorů 2. a 3. kloubu standardního manipulátoru. Takový koncept kinematické architektury manipulátoru lze v poslední době pozorovat u některých velkých robotických firem, např. robot LBR iiwa společnosti Kuka². Přesto, že takové koncepty manipulátorů jsou využívány v průmyslových aplikacích zatím pouze velmi omezeně, lze vyzdvihnout některé zásadní výhody, které motivují jejich další vývoj a výzkum.

- **Zásadní navýšení možností dosahu robotu - zvětšení pracovního prostoru**

V důsledku navýšení počtu stupňů volnosti robotu lze při zachování fyzického pracovního prostoru robotu (minimální zástavbový prostor, ve kterém se robotu může pohybovat) vykonávat s robotem pohyby, které zásadním způsobem zvyšují dosah koncového efektoru do míst, které jsou pro standardní průmyslový robot stejné velikosti obtížně dosažitelné, či nedosažitelné vůbec.

- **Možné optimalizace pohybu manipulátoru**

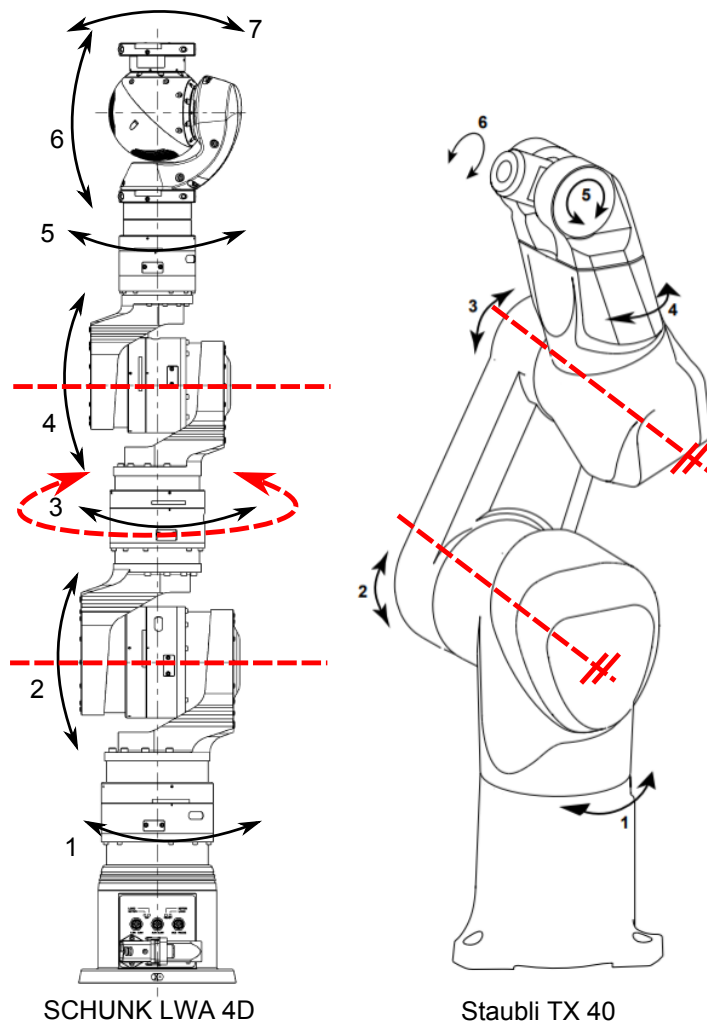
Vzhledem ke kinematické redundanci manipulátoru lze tuto vlastnost efektivně využít k řízení pohybu manipulátoru po konkrétní zadané trajektorii při současné optimalizaci (minimalizaci/maximalizaci) zvoleného kritéria (např. vzdálenosti od singulárních poloh, minimalizace rychlostí aktuátorů či omezení možnosti, že aktuátory budou dosahovat svých limitních poloh).

Samozřejmě výše uvedené výhody zefektivňující pohyb manipulátoru s sebou přinášejí i dílčí nevýhody, které zpomalují proces integrace takových robotů do výroby. Jde především o následující:

- Nestandardní architektury manipulátorů v porovnání s průmyslovými roboty
- Nutnost osazovat další aktuátory
- Komplikovanější řídicí systém manipulátoru s ohledem na jeho vývoj, implementaci a zajištění spolehlivosti
- Integrace nestandardních funkcí (včetně pokročilých možností konfigurace) s ohledem na operátorské rozhraní

¹Jednotka FTM 45 je výhradně dodávána k manipulátoru LWA 4D a podrobné informace jsou dodávány s dokumentací k samotnému robotu.

²<https://www.kuka.com/cs-cz/produkty,-sluzby/roboticke-systemy/prumyslove-roboty/lbriiwa>



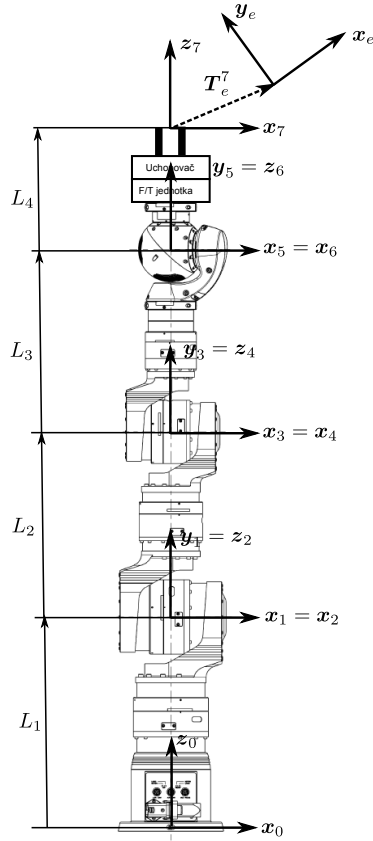
Obrázek 1: Umístění os rotace aktuátorů standardního průmyslového robotu Staubli TX 40 a uvažovaného manipulátoru SCHUNK LWA 4D.

2 Virtuální simulační model manipulátoru

Virtuální simulační model manipulátoru byl vytvořen procesem matematicko-fyzikálního modelování a implementován v programovém prostředí Matlab/Simulink/SimMechanics [5]. Pro tvorbu simulačního modelu byly využity předimplemtnované funkce v Matlabu a funkční bloky v prostředí Simulink/SimMechanics, které jsou součástí knihovny robotLib [9].

Kinematický model manipulátoru je realizován standardním postupem přiřazení souřadných systémů (s.s.) dílčím ramenům manipulátoru dle Denavit-Hartenbergovy (D-H) úmluvy [1], viz Obrázek 2.

D-H parametry jsou shrnuty v Tabulce 1.



Obrázek 2: S.s. jednotlivých ramen manipulátoru dle D-H úmluvy

i	d_i	θ_i	a_i	α_i
1	L_1	θ_1	0	$\frac{\pi}{2}$
2	0	θ_2	0	$-\frac{\pi}{2}$
3	L_2	θ_3	0	$\frac{\pi}{2}$
4	0	θ_4	0	$-\frac{\pi}{2}$
5	L_3	θ_5	0	$\frac{\pi}{2}$
6	0	θ_6	0	$-\frac{\pi}{2}$
7	L_4	θ_7	0	0

Tabulka 1: DH parametry manipulátoru

Kloubové souřadnice manipulátoru, tedy poloha rotačních aktuátorů, jsou dány jako:

$$\mathbf{Q} = [\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7]^T \quad (1)$$

Zobecněné souřadnice manipulátoru, tedy poloha (translace a orientace) koncového efektoru, jsou dány jako³ :

$$\mathbf{X} = [\mathbf{O}_e^0 \quad \mathbf{R}_e^0] \quad (2)$$

kde \mathbf{O}_e^0 je počátek s.s. systému koncového efektoru F_e a \mathbf{R}_e^0 reprezentuje jeho orientaci v podobě matice rotace (obojí vztaheno k s.s. F_0).

³Značení bodu/vektoru/matice \mathbf{x}_i^j reprezentuje bod/vektor/matice \mathbf{x}_i jejíž souřadnice (prvky) jsou vyjádřeny vzhledem k s.s. F_j .

Kinematické návrhové parametry manipulátoru jsou dány jako:

$$\boldsymbol{\xi} = [L_1 \ L_2 \ L_3 \ L_4]^T \quad (\text{délky ramen}) \quad (3)$$

$$\mathbf{R}_e^7, \mathbf{O}_e^7 \quad (\text{translace a orientace konc. efektoru v s.s. } F_7, \text{ kompenzace}) \quad (4)$$

2.1 Přímá kinematická úloha

Přímá kinematická úloha je dána standardním přístupem skládání transformací mezi zavedenými s.s. reprezentované postupným násobením homogenních transformačních matic⁴ \mathbf{T}_i^{i-1} :

$$\mathbf{T}_e^0 = \prod_{i=1}^7 \mathbf{T}_i^{i-1}(d_i, \theta_i, a_i, \alpha_i) \cdot \mathbf{T}_e^7, \quad \mathbf{T}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_e^7 = \begin{bmatrix} \mathbf{R}_e^7 & \mathbf{O}_e^7 \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad \mathbf{O}_e^0 = \mathbf{T}_e^0[1:3, 4], \quad \mathbf{R}_e^0 = \mathbf{T}_e^0[1:3, 1:3] \quad (5)$$

kde $d_i, \theta_i, a_i, \alpha_i$ jsou D-H parametry, viz Tabulka 1, včetně odpovídajících kloubových souřadnic.

Okamžitá přímá kinematická úloha (závislosti rychlostí a zrychlení) lze odvodit ze známého vztahu mezi rychlostmi zobecněných a kloubových souřadnic:

$$\begin{aligned} \dot{\mathbf{X}} &= \begin{bmatrix} \dot{\omega}_e^0 \\ \dot{\omega}_e^0 \end{bmatrix} = \underbrace{\mathbf{J}_{\text{comp}}(\mathbf{Q}) \cdot \mathbf{J}_7^0(\mathbf{Q})}_{=\mathbf{J}(\mathbf{Q})} \cdot \dot{\mathbf{Q}} \\ \ddot{\mathbf{X}} &= \begin{bmatrix} \ddot{\omega}_e^0 \\ \ddot{\omega}_e^0 \end{bmatrix} = \underbrace{\mathbf{J}_{\text{comp}}(\mathbf{Q}) \cdot \dot{\mathbf{J}}_7^0(\mathbf{Q}, \dot{\mathbf{Q}})}_{=\mathbf{J}(\mathbf{Q}, \dot{\mathbf{Q}}) \cdot \dot{\mathbf{Q}}} + \underbrace{\mathbf{J}_{\text{comp}}(\mathbf{Q}) \cdot \mathbf{J}_7^0(\mathbf{Q})}_{\mathbf{J}(\mathbf{Q})} \cdot \ddot{\mathbf{Q}} \end{aligned} \quad (6)$$

kde $\mathbf{J}_7^0(\mathbf{Q})$ resp. $\dot{\mathbf{J}}_7^0(\dot{\mathbf{Q}}, \mathbf{Q})$ je kinematický jakobián manipulátoru (bez uvažování kompenzací) resp. jeho časová derivace, která lze systematicky odvodit přímo z hodnot prvků homogenních transformačních matic \mathbf{T}_i^{i-1} , $i = 1 \dots 7$ a \mathbf{J}_{comp} resp. \mathbf{J}_{add} je kompenzační matice resp. aditivní vektor zahrnující vliv kompenzace polohy koncového efektoru maticí \mathbf{T}_e^7 . ω_e^0 resp. $\dot{\omega}_e^0$ je vektor úhlové rychlosti resp. úhlového zrychlení. Efektivní algoritmy pro výpočet lze nalézt v [7] a jsou součástí knihovny robotLib. \mathbf{J} je výsledný kinematický jakobián manipulátoru (s uvažováním kompenzací).

2.2 Inverzní kinematická úloha

Inverzní kinematika, tedy stanovení polohy kloubů \mathbf{Q} ze známé polohy koncového efektoru \mathbf{X} , je pro redundantní manipulátory úloha s nekonečným počtem řešení. Konkrétní řešení je poté parametrizováno vektorem dimenze $n - m$ tzv. redundantních kloubových souřadnic, kde m je počet DoF koncového efektoru manipulátoru a n je počet nezávislých kloubových souřadnic manipulátoru. Volba redundantních kloubových souřadnic z vektoru kloubových souřadnic \mathbf{Q} může být provedena libovolně. V našem případě se, s ohledem na paralelu ke kinematice standardních manipulátorů, zdá rozumné volit jako redundantní kloubovou souřadnici θ_3 (polohu 3. osy), která poté tvoří společně s požadovanou polohou koncového efektoru \mathbf{X} vstup algoritmu inverzní kinematické úlohy, viz Algoritmus 1.

⁴Označení $\mathbf{O}[a : b, c : d]$ je výběr a -tého až b -tého řádku a c -tého až d -tého sloupce matice/vektoru \mathbf{O} . Označení $\mathbf{O}[:, c : d]$ je výběr všech řádků a c -tého až d -tého sloupce. $s_{\theta_i} = \sin(\theta_i)$, $c_{\theta_i} = \cos(\theta_i)$

◆ **Algoritmus 1 (Inverzní kinematická úloha)**

Vstup:

Poloha, rychlost, zrychlení zobecněných souřadnic

$$\mathbf{X} = [\mathbf{O}_e^0 \quad \mathbf{R}_e^0], \quad \dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{O}}_e^0 \\ \dot{\boldsymbol{\omega}}_e^0 \end{bmatrix}, \quad \ddot{\mathbf{X}} = \begin{bmatrix} \ddot{\mathbf{O}}_e^0 \\ \ddot{\boldsymbol{\omega}}_e^0 \end{bmatrix}$$

Poloha, rychlost, zrychlení redundantní kloubové souřadnice (parametrizace):

$$\mathbf{Q}_{\text{par}} = \theta_3, \quad \dot{\mathbf{Q}}_{\text{par}} = \dot{\theta}_3, \quad \ddot{\mathbf{Q}}_{\text{par}} = \ddot{\theta}_3$$

Vybrané řešení inverzní kinematické úlohy:

$$\text{sol} = [i, j, k], \quad i, j, k \in \{1, 2\} \quad (7)$$

Výstup:

Polohy, rychlost, zrychlení zbývajících kloubových souřadnic:

$$\begin{aligned} \mathbf{Q}_{\text{orig}} &= [\theta_1 \quad \theta_2 \quad \theta_4 \quad \theta_5 \quad \theta_6 \quad \theta_7]^T \\ \dot{\mathbf{Q}}_{\text{orig}} &= [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_4 \quad \dot{\theta}_5 \quad \dot{\theta}_6 \quad \dot{\theta}_7]^T \\ \ddot{\mathbf{Q}}_{\text{orig}} &= [\ddot{\theta}_1 \quad \ddot{\theta}_2 \quad \ddot{\theta}_4 \quad \ddot{\theta}_5 \quad \ddot{\theta}_6 \quad \ddot{\theta}_7]^T \end{aligned}$$

Algoritmus pro závislosti POLOHY:

Výsledný algoritmus je uveden v Příloze B jako realizace v Matlabu, kde platí následující substituce:

$$\text{O70} = \mathbf{T}_7^0[1 : 3, 4], \quad \text{R70} = \mathbf{T}_7^0[1 : 3, 1 : 3]$$

kde \mathbf{T}_7^0 je transformační matice s.s. posledního ramene manipulátoru (bez kompenzace polohy koncového efektoru):

$$\mathbf{T}_7^0 = \begin{bmatrix} \mathbf{R}_e^0 & \mathbf{O}_e^0 \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_e^7 & \mathbf{O}_e^7 \\ \mathbf{0} & 1 \end{bmatrix}^{-1}$$

$$\text{L1} = L_1, \text{L2} = L_2, \text{L3} = L_3, \text{L4} = L_4, \text{sol} = \text{sol}, \text{theta3} = \theta_3$$

$$\mathbf{Q}_{\text{orig}} = \mathbf{Q}[1, 2, 4 : 7]$$

Algoritmus pro závislosti RYCHLOSTI, ZRYCHLENÍ:

Vyjdeme-li ze vztahů pro přímou okamžitou kinematickou úlohu (6), lze s využitím zavedené redundantní souřadnice θ_3 psát vztahy pro rychlosti a zrychlení zbývajících kloubových souřadnic následovně:

$$\dot{\mathbf{Q}}_{\text{orig}} = \mathbf{J}_{\text{orig}}^{-1} \cdot (\dot{\mathbf{X}} - \mathbf{J}_{\text{par}} \cdot \dot{\mathbf{Q}}_{\text{par}}) \quad (8)$$

$$\ddot{\mathbf{Q}}_{\text{orig}} = \mathbf{J}_{\text{orig}}^{-1} \cdot (\ddot{\mathbf{X}} - \dot{\mathbf{J}} \cdot \dot{\mathbf{Q}} - \mathbf{J}_{\text{par}} \cdot \ddot{\mathbf{Q}}_{\text{par}}) \quad (9)$$

kde pro jakobiány $\mathbf{J}_{\text{orig}} = \mathbf{J}(\mathbf{Q})[:, [1, 2, 4 : 7]]$ a $\mathbf{J}_{\text{par}} = \mathbf{J}(\mathbf{Q})[:, 3]$ platí:

$$\dot{\mathbf{X}} = \mathbf{J}_{\text{orig}} \cdot \dot{\mathbf{Q}}_{\text{orig}} + \mathbf{J}_{\text{par}} \cdot \dot{\mathbf{Q}}_{\text{par}}$$

◆

2.3 Dynamika manipulátoru

Přímý a inverzní dynamický model manipulátoru lze v případě znalosti výše uvedených kinematických výpočtů efektivně implementovat v podobě rekurzivních algoritmů pro výpočet distribuce rychlostí/zrychlení dílčích ramen manipulátorů (tedy příslušných s.s.) od základny manipulátoru ke koncovému efektoru (dopředná rekurze) a distribuce sil/momentů působící na ramena od koncového efektoru do základny manipulátoru (zpětná rekurze). Výsledkem je inverzní dynamický model manipulátoru umožňující výpočet požadovaných sil/momentů aktuátorů na základě znalosti požadovaného pohybu koncového efektoru (poloha, rychlost, zrychlení) a externích sil/momentů působící na koncový efektor. Přímý dynamický model (výpočet pohybu manipulátoru na základě působících sil/momentů v aktuátorech) lze získat opět rekurzivním algoritmem využívající inverzní dynamický model. Detailní informace lze nalézt v kapitole *A.6. Automatické generování dynamických modelů* v publikaci [9].

Dynamické modely manipulátoru lze zpravidla efektivně využít k návrhu pokročilých (např. decentralizovaných) metod řízení manipulátorů, viz [2, 3], či jako podpurné prostředky pro výpočet účelové funkce při optimalizaci robotických architektur, viz [9]. V případě výzkumu algoritmů řízení pro manipulátor SCHUNK LWA 4D však žádná z těchto možností nebyla využita, neboť pohony uvažovaného robotu jsou tvořeny kompaktními aktuátory obsahující vlastní regulační smyčku jejíž vstupem je přímo setpoint na požadovanou polohu (možná konfigurace regulátoru uživatelem je silně omezena). Vzhledem k dané kinematice manipulátoru (konfigurace kloubů, délky ramen) nebyly algoritmy kinematické optimalizace manipulátoru v našem případě relevantní.

3 Optimální řízení pohybu robotu

Redundanci manipulátoru, která s sebou přináší možnost parametrizace inverzní kinematické úlohy redundantní kloubovou souřadnicí θ_3 , lze využít k optimalizaci pohybu manipulátoru. Vzhledem k cílovému nasazení manipulátoru pro obsluhu zařízení v biolaboratoři, kde manipulátor přesouval zásobníky se vzorky (tzv. „plate“) mezi danými pozicemi určených zařízení), viz [6], byl proces plánování (učení) pohybu manipulátoru volen následovně:

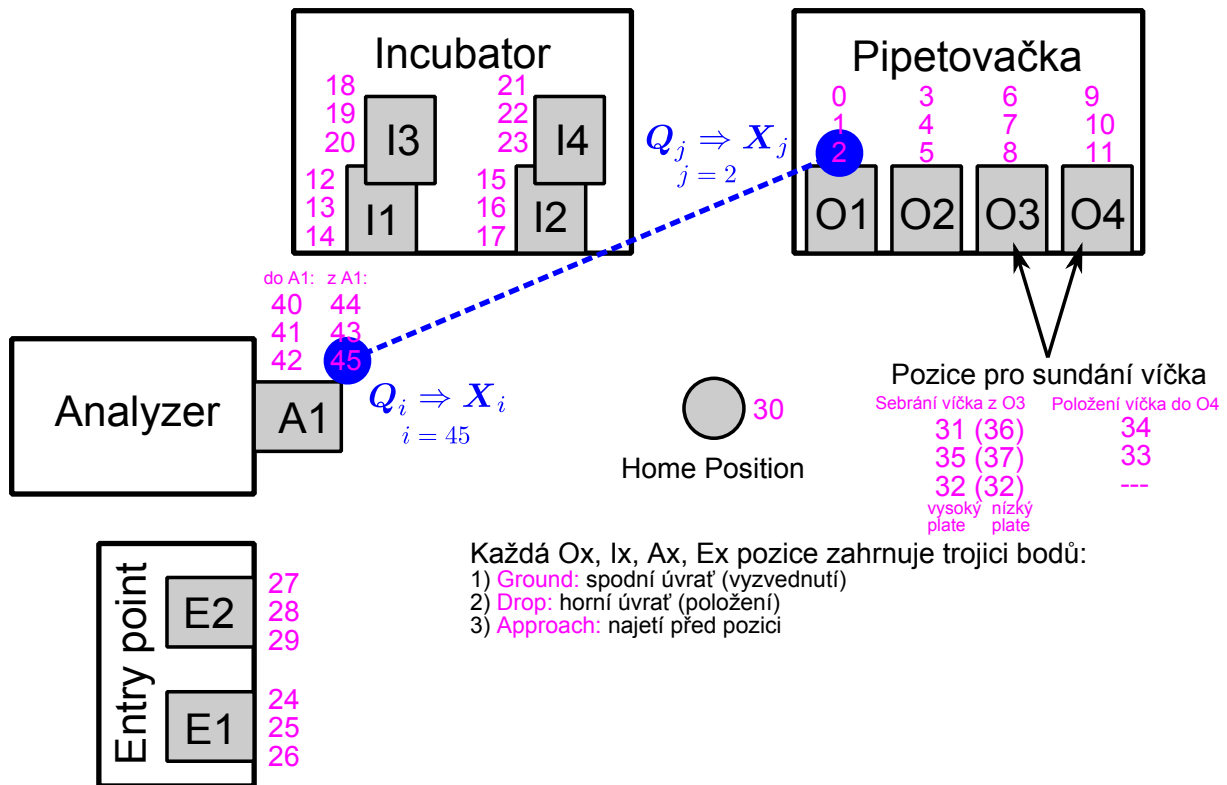
- **Učení počátečních a koncových bodů dílčích trajektorií:**

V režimu koordinovaného JOGu (JOG⁵ v aktuálních souřadnicích koncového efektoru manipulátoru) je naučena počáteční \mathbf{X}_i a cílová \mathbf{X}_j poloha dílčího pohybu koncového efektoru a zaznamenána jako dvojice kloubových souřadnic manipulátoru $\mathbf{Q}_i, \mathbf{Q}_j$, viz Obrázek 3. Poznamenejme, že záznam v prostoru kloubových souřadnic principiálně vylučuje možnost chyby v případě najetí do odpovídajících zobecněných souřadnic v jiné konfiguraci (řešení sol či jinou hodnotou redundantní kloubové souřadnice θ_3) manipulátoru. Tento problém vznikal při reálném nasazení manipulátoru pravděpodobně vlivem nepřesností v kinematice manipulátoru (kterou nebylo možné přesněji zkalibrovat).

- **Plánování přímkového pohybu mezi naučenými body trajektorie:**

Automatický režim, který pro dvojici zadaných kloubových souřadnic $\mathbf{Q}_i, \mathbf{Q}_j$ naplánoval přímkový pohyb v prostoru souřadnic zobecněných $\mathbf{X}_i + k \cdot (\mathbf{X}_j - \mathbf{X}_i)$, $k \in \langle 0, 1 \rangle$ s definovaným profilem požadované maximální rychlosti a zrychlení (lichoběžníkový profil rychlosti), viz [8]. Zároveň bylo algoritmem zajištěno, aby počáteční i koncový stav manipulátoru korespondoval s naučenými body $\mathbf{Q}_i, \mathbf{Q}_j$ (viz výše uvedený důvod vyplývající z kalibrace návrhových parametrů kinematického modelu).

⁵JOG v robotice znamená ruční pojezd robotu v prostoru kloubových souřadnic (nekoordinovaně, tzv. „JOG in JOINTS“) či v prostoru zobecněných souřadnic (koordinovaně, tzv. „JOG in WORLD, JOG in TOOL“)



Obrázek 3: Adresy stanovišť v reálné aplikaci, komplexní trajektorie jsou dány složením dílčích přímkových trajektorií z klidu do klidu (bez blendingu). Optimální plánování pohybu trajektorie manipulatoru probíhá v rámci zvolených dílčích přímkových úseků.

3.1 Učení počátečních a koncových bodů dílčích trajektorií

Triviální možností koordinovaného JOGu v aktuálním s.s. F_e koncového efektoru je manuální dostavování redundantní kloubové souřadnice θ_3 podél pohybu koncového efektoru, tzn. vedle požadované rychlosti translace \dot{O}_e^0 a orientace ω_e^0 diktovat i požadovanou rychlost $\dot{\theta}_3$.

Bohužel takový přístup není v podstatě možný, neboť není vůbec intuitivně zřejmé, jakým způsobem manuálně řídit rychlost redundantní kloubové souřadnice θ_3 . Generování rychlosti kloubových souřadnic $Q = \{Q_{\text{orig}}, Q_{\text{par}}\}$ tak musí být řešeno jako optimalizační úloha. V případě uvažovaného manipulatoru SCHUNK LWA 4D byly pozorovány následující kritické požadavky na optimalizaci pohybu během JOGu:

- Relativně striktní omezení na rozsahy pohybu kloubů robotu zásadně limitují pracovní prostor \Rightarrow **nutnost držet co nejvíce polohy kloubů uprostřed povolených rozsahů**
- **Omezovat kloubovým souřadnicím dosahovat vysoké rychlosti a v případě překročení povolených rozsahů tyto rychlosti limitovat** na úkor rychlosti koordinovaného pohybu

Algoritmus výpočtu optimálních kloubových rychlostí je inspirován známým přístupem k řízení pohybu redundantních manipulatorů [4, 3, 9] a lze formulovat následovně:

$$\dot{Q} = J^\dagger(Q) \cdot \dot{X} + (I - J^\dagger(Q) \cdot J(Q)) \cdot \frac{\partial w(Q)}{\partial Q} \quad (10)$$

kde $\mathbf{J}^\dagger(\mathbf{Q}) = \mathbf{J}^T(\mathbf{Q}) \cdot (\mathbf{J}(\mathbf{Q}) \cdot \mathbf{J}^T(\mathbf{Q}))^{-1}$ je zobecněná inverze kinematického jakobiánu $\mathbf{J}(\mathbf{Q})$, viz (6),

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{O}}_e^0 \\ \boldsymbol{\omega}_e^0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_e^0(\mathbf{Q}) \cdot \dot{\mathbf{O}}_e^e \\ \mathbf{R}_e^0(\mathbf{Q}) \cdot \boldsymbol{\omega}_e^e \end{bmatrix}$$

je požadovaná rychlost koncového efektoru (koordinovaná JOGovací rychlost) v s.s. F_0 (základny) a $\dot{\mathbf{O}}_e^e$ resp. $\boldsymbol{\omega}_e^e$ je (operátorem) zadaná JOGovací rychlost koncového efektoru manipulátoru vzhledem k aktuálnímu s.s. koncového efektoru F_e (v robotice známo jako „JOG in Tool Frame“).

Kriteriální funkce $w(\mathbf{Q})$ byla volena jako:

$$w(\mathbf{Q}) = -k_0 \cdot \frac{1}{2n} \sum_{i=1}^n \left(\frac{\theta_i - \bar{\theta}_i}{\theta_i^{max} - \theta_i^{min}} \right)^2, \quad n = 7$$

$$\mathbf{Q} = [\theta_1, \theta_2, \dots, \theta_n]^T \quad (11)$$

$$\mathbf{Q}^{min} = [\theta_1^{min}, \theta_2^{min}, \dots, \theta_n^{min}]^T$$

$$\mathbf{Q}^{max} = [\theta_1^{max}, \theta_2^{max}, \dots, \theta_n^{max}]^T$$

kde $\bar{\theta}_i = \theta_i^{min} + \frac{\theta_i^{max} - \theta_i^{min}}{2}$ a θ_i^{max} resp. θ_i^{min} označuje maximální resp. minimální polohu kloubu. Maximalizací kriteriální funkce rekonfigurujeme mechaniku manipulátoru takovým způsobem, aby se klouby manipulátoru udržovaly co nejdále od svých limitních poloh. Konstanta k_0 odpovídá realizuje kompromis mezi maximalizací vzdáleností od singularních poloh manipulátoru ($k_0 \gg 0$) a minimalizací kloubových rychlostí ($k_0 \rightarrow 0$).

Gradient kriteriální funkce v (10) je poté dán následovně:

$$\frac{\partial w(\mathbf{Q})}{\partial \mathbf{Q}} = -k_0 \cdot \frac{1}{n} \left[\frac{\theta_1 - \bar{\theta}_1}{(\theta_1^{max} - \theta_1^{min})^2}, \frac{\theta_2 - \bar{\theta}_2}{(\theta_2^{max} - \theta_2^{min})^2}, \dots, \frac{\theta_n - \bar{\theta}_n}{(\theta_n^{max} - \theta_n^{min})^2} \right]^T \quad (12)$$

Algoritmus byl dále doplněn o systém redukce maximálních možných rychlostí kloubů, který zajišťoval zpomalení celkového koordinovaného pohybu v případě, že některá kloubová rychlost přesáhne dovolenou mez. Takový bezpečnostní prvek umožňuje operátorovi být zpomalený, ale stále koordinovaný pohyb robotu i v oblastech blízkých singularním polohám, kdy by za normálních situacích (bez redukce) docházelo k neúnosnému zvýšení požadovaných kloubových rychlostí (a tedy rychlé a potenciálně nebezpečné rekonfiguraci ramen manipulátoru). Algoritmus lze stručně popsat následovně.

♦ Algoritmus 2 (Optimální koordinovaný JOG)

1. Vypočti optimální hodnotu kloubových rychlostí $\dot{\mathbf{Q}}$ dle vztahu (10)

2. Pokud:

$$\max_i (\|\dot{\mathbf{Q}}[i]\|) > \dot{\theta}^{max} \quad (\text{Překročena max. povolená rychlost kloubů})$$

3. Uprav hodnotu kloubových rychlostí následovně:

$$\dot{\mathbf{Q}} = k_{\text{fact}} \cdot \dot{\mathbf{Q}}, \quad k_{\text{fact}} = \frac{\dot{\theta}^{max}}{\max_i (\|\dot{\mathbf{Q}}[i]\|)} < 1$$

lze snadno ukázat, že skalární konstanta k_{fact} zpomalí výsledný pohyb koncového efektoru, ale nezmění jeho směr (koordinovaný pohyb zůstane zachován).

4. Pokud k_{fact} je příliš malé (manipulátor je velmi blízko singularity a pohyb koncového efektoru generovaný max. možnou rychlostí kloubů už by byl příliš pomalý) zastav možnost JOGování.

♦

Vypočtená optimální hodnota rychlostí kloubových souřadnic $\dot{\mathbf{Q}}$ byla následně využita v implementovaném řídicím systému, viz [6], jako setpoint rychlostních (po integraci polohových) regulátorů dílčích kloubů.

S uvedeným algoritmem optimálního JOGu robotu v aktuálním s.s. koncového efektoru je pak operátor velmi intuitivním způsobem schopen navést manipulátor do požadovaných pozic pro naučení počátečních a koncových bodů dílčích trajektorií.

3.2 Plánování přímkového pohybu mezi naučenými body trajektorie

V okamžiku, kdy jsou naučeny všechny zájmové (koincidenční body), je nutné generovat mezi těmito body přímkové pohyby s konstantní orientací v osách $\mathbf{x}_0\mathbf{y}_0$ (zásobník se vzorky se nesmí naklánět). Současně požadujeme, aby během takového koordinovaného pohybu robotu byl pohyb optimalizován dle stejného kritéria (11, 12) jako v případě optimálního JOGu. Takový požadavek lze snadno řešit rozšířením algoritmu (10) pro generování optimálních kloubových rychlostí o člen vyčísující polohovou odchylku mezi požadovanou a skutečnou (vypočítanou dle aktuálních hodnot kloubových poloh) polohou koncového efektoru [9]:

$$\begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_o \end{bmatrix} = \mathbf{e} = (\mathbf{X}_d - \mathbf{G}(\mathbf{Q})) = \begin{bmatrix} \mathbf{O}_d - \mathbf{O} \\ \text{Jak je definováno?} \end{bmatrix} \quad (13)$$

$$\begin{bmatrix} \dot{\mathbf{e}}_p \\ \dot{\mathbf{e}}_o \end{bmatrix} = \dot{\mathbf{e}} = (\dot{\mathbf{X}}_d - \mathbf{J}(\mathbf{Q}) \cdot \dot{\mathbf{Q}}) = \begin{bmatrix} \dot{\mathbf{O}}_d - \dot{\mathbf{O}} \\ \boldsymbol{\omega}_d - \boldsymbol{\omega} \end{bmatrix} \quad (14)$$

kde \mathbf{e}_p resp \mathbf{e}_o je chyba v translaci resp. orientaci koncového efektoru, funkce $\mathbf{G}(\mathbf{Q})$ reprezentuje přímou kinematickou úlohu pro polohy (5). Výsledný vztah pro výpočet optimálních kloubových rychlostí

$$\dot{\mathbf{Q}} = \mathbf{J}^\dagger(\mathbf{Q}) \cdot \left(\dot{\mathbf{X}}_d + \mathbf{K}(\mathbf{X}_d - \mathbf{G}(\mathbf{Q})) \right) + \left(\mathbf{I} - \mathbf{J}^\dagger(\mathbf{Q}) \cdot \mathbf{J}(\mathbf{Q}) \right) \cdot \frac{\partial w(\mathbf{Q})}{\partial \mathbf{Q}} \quad (15)$$

kde matice

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_o \end{bmatrix}$$

je diagonální pozitivně definitní matice zesílení, pak vede na diferenciální rovnici generující časový vývoj chyby (důkaz spočívá v přenásobení rovnice (15) jakobiánem \mathbf{J} zleva):

$$\dot{\mathbf{e}} + \mathbf{K} \cdot \mathbf{e} = \mathbf{0} \Rightarrow \begin{bmatrix} \dot{\mathbf{e}}_p \\ \dot{\mathbf{e}}_o \end{bmatrix} + \begin{bmatrix} \mathbf{K}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_o \end{bmatrix} \cdot \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_o \end{bmatrix} = \mathbf{0} \quad (16)$$

Chybu v poloze koncového efektoru lze jednoduše definovat pro translaci přímo rozdílem mezi skutečnou xyz translací \mathbf{O} a vypočtenou xyz translací \mathbf{O}_d koncového efektoru:

$$\begin{aligned} \mathbf{e}_p &= \mathbf{O}_d - \mathbf{O}, & \mathbf{O} &= \mathbf{O}_e^0(\mathbf{Q}) \text{ (z řešení přímé kinematiky pro polohy (5))} \\ \dot{\mathbf{e}}_p &= \dot{\mathbf{O}}_d - \dot{\mathbf{O}}, & \dot{\mathbf{O}} &= \dot{\mathbf{O}}_e^0(\mathbf{Q}, \dot{\mathbf{Q}}) \text{ (z řešení přímé kinematiky pro rychlosti (6))} \end{aligned} \quad (17)$$

Tedy jistě platí $\dot{\mathbf{e}}_p = \frac{d}{dt}\mathbf{e}_p$, diferenciální rovnice chyby $\dot{\mathbf{e}}_p + \mathbf{K}_p \cdot \mathbf{e}_p = \mathbf{0}$ je lineární a pro $\mathbf{K}_p = \mathbf{K}[1 : 3, 1 : 3]$ pozitivně definitní tedy stabilní a konvergující k nulové chybě v translaci.

Otázkou zůstává, jak definovat chybu v orientaci \mathbf{e}_o . Nabízí se např. možnost ji v minimální reprezentaci definovat jako rozdíl skutečných a požadovaných Eulerových úhlů. Zde však nastává zásadní problém s tzv. singularitou v reprezentaci [9], kde nelze jednoznačně převést při řešení přímé kinematiky pro polohy získanou matici rotace (konkrétně \mathbf{R}_e^0) na Eulerovy úhly (existují dvě možná řešení které jsou izolována singularitou, tzv. „gimbal lock“). Tento fakt definici chyby

v orientaci zásadně komplikuje. Řešením je využít definici chyby v orientaci prostřednictvím tzv. jednotkového kvaternionu $\mathbf{Quat} \in \mathbb{R}^4$ [3, 2], kde existuje jednoznačný vztah mezi maticí rotací a jednotkovým kvaternionem:

$$\mathbf{Quat} = [\vartheta \quad \boldsymbol{\epsilon}^T]^T = [\vartheta \quad \epsilon_x \quad \epsilon_y \quad \epsilon_z]^T \quad (18)$$

Chybu v orientaci mezi požadovanou orientací vyjádřenou kvaternionem \mathbf{Quat}_d a vypočtenou orientací $\mathbf{Quat} \leftarrow \mathbf{R}_e^0(\mathbf{Q})$ (z řešení přímé kinematiky pro polohy (5)) lze definovat následovně:

$$\mathbf{e}_o = \vartheta \cdot \boldsymbol{\epsilon}_d - \vartheta_d \cdot \boldsymbol{\epsilon} - \mathbf{S}(\boldsymbol{\epsilon}_d) \cdot \boldsymbol{\epsilon} \quad (19)$$

kde $\mathbf{S}(\boldsymbol{\epsilon})$ je antisymetrická matice

$$\mathbf{S}(\boldsymbol{\epsilon}) = \begin{bmatrix} 0 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 0 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 0 \end{bmatrix}$$

Chybu mezi požadovanou $\boldsymbol{\omega}_d$ a vypočtenou úhlovou rychlostí $\boldsymbol{\omega} = \boldsymbol{\omega}_e^0(\mathbf{Q}, \dot{\mathbf{Q}})$ (z řešení přímé kinematiky pro rychlosti (6)) lze psát přímo jako $\boldsymbol{\omega}_d - \boldsymbol{\omega}$, dostáváme tak diferenciální rovnici vývoje chyby v orientaci:

$$\boldsymbol{\omega}_d - \boldsymbol{\omega} + \mathbf{K}_o \cdot \mathbf{e}_o = \mathbf{0}, \quad \mathbf{K}_o = \mathbf{K}[4 : 6, 4 : 6] \quad (20)$$

Bohužel, diferenciální rovnice (20) již není lineární, neboť $\boldsymbol{\omega}_d - \boldsymbol{\omega} \neq \frac{d}{dt}\mathbf{e}_o$. Ale zavedením kandidáta na Ljapunovovu funkci ve tvaru

$$\begin{aligned} V(\mathbf{e}_o) &= (\vartheta_d - \vartheta)^2 + (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon})^T \cdot (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}) \geq 0 \\ \dot{V}(\mathbf{e}_o, \dot{\mathbf{e}}_o) &= 2(\vartheta_d - \vartheta) \cdot (\dot{\vartheta}_d - \dot{\vartheta}) + 2(\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon})^T \cdot (\dot{\boldsymbol{\epsilon}}_d - \dot{\boldsymbol{\epsilon}}) \end{aligned} \quad (21)$$

a s využitím vztahu mezi úhlovou rychlostí a časovou derivací kvaternionu (kvaternion propagation)

$$\dot{\vartheta} = -\frac{1}{2}\boldsymbol{\epsilon}^T \cdot \boldsymbol{\omega}, \quad \dot{\boldsymbol{\epsilon}} = \frac{1}{2}(\vartheta\mathbf{I} - \mathbf{S}(\boldsymbol{\omega})) \cdot \boldsymbol{\omega} \quad (22)$$

lze ukázat, že:

$$\dot{V} = -\mathbf{e}_o \cdot \mathbf{K}_o \cdot \mathbf{e}_o < 0 \quad (23)$$

A tedy V je Ljapunovovou funkcí a diferenciální rovnice (20) je stabilní pro \mathbf{K}_o pozitivně definitní a konvergující k nulové chybě v orientaci.

Výpočet optimálních kloubových rychlostí (15) tedy přejde na konkrétní výpočetní tvar:

$$\dot{\mathbf{Q}} = \mathbf{J}^\dagger(\mathbf{Q}) \cdot \left(\dot{\mathbf{X}}_d + \begin{bmatrix} k_p \cdot \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_o \cdot \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_o \end{bmatrix} \right) + \left(\mathbf{I} - \mathbf{J}^\dagger(\mathbf{Q}) \cdot \mathbf{J}(\mathbf{Q}) \right) \cdot \frac{\partial w(\mathbf{Q})}{\partial \mathbf{Q}} \quad (24)$$

kde k_p, k_o jsou skalární konstanty zesílení ($\Rightarrow \mathbf{K}_p, \mathbf{K}_o$ jsou diagonální pozitivně definitní matice se stejnými prvky na diagonále), $w(\mathbf{Q})$ je kriteriální funkce identická s (11) a požadované a vypočtené polohy a rychlosti koncového efektoru a příslušné chyby v poloze a orientaci jsou dány jako:

$$\text{Požadovaná poloha konc. eff.:} \quad \mathbf{X}_d = \begin{bmatrix} \mathbf{O}_d \\ \mathbf{Quat}_d \end{bmatrix}, \quad \mathbf{Quat}_d = [\vartheta_d \quad \boldsymbol{\epsilon}_d^T]^T$$

$$\text{Požadovaná rychlost konc. eff.:} \quad \dot{\mathbf{X}}_d = \begin{bmatrix} \dot{\mathbf{O}}_d \\ \boldsymbol{\omega}_d \end{bmatrix}$$

$$\text{Vypočtená poloha konc. eff.:} \quad \begin{bmatrix} \mathbf{O} \\ \mathbf{Quat} \end{bmatrix} = \begin{bmatrix} \mathbf{O}_e^0(\mathbf{Q}) \\ \mathbf{Quat}_e^0(\mathbf{Q}) \leftarrow \mathbf{R}_e^0(\mathbf{Q}) \end{bmatrix}, \quad \mathbf{Quat} = [\vartheta \quad \boldsymbol{\epsilon}^T]^T$$

$$\text{Chyba polohy v translaci:} \quad \mathbf{e}_p = \mathbf{O}_d - \mathbf{O}$$

$$\text{Chyba polohy v orientaci:} \quad \mathbf{e}_o = \vartheta \cdot \boldsymbol{\epsilon}_d - \vartheta_d \cdot \boldsymbol{\epsilon} - \mathbf{S}(\boldsymbol{\epsilon}_d) \cdot \boldsymbol{\epsilon}$$

◆ **Algoritmus 3 (Generátor opt. kloubové trajektorie pro přímkový pohyb)**

Následující algoritmus schématicky (bez podrobnějších implementačních poznámek) shrnuje princip generování kloubových souřadnic manipulátoru zajišťující přímkový přesun koncového efektoru manipulátoru mezi dvěma naučenými body v pracovním prostoru.

1. Z dvojice naučených bodů v pracovním prostoru manipulátoru (počáteční a koncový bod přímkového úseku) $\mathbf{Q}_i, \mathbf{Q}_j$ lze vygenerovat požadovaný pohyb koncového efektoru jako⁶:

$$\begin{aligned}\mathbf{X} &= \mathbf{X}_i + k \cdot (\mathbf{X}_j - \mathbf{X}_i), \quad \mathbf{X}_i = \mathbf{G}(\mathbf{Q}_i), \quad \mathbf{X}_j = \mathbf{G}(\mathbf{Q}_j), \quad k = k(t) \in \langle 0, 1 \rangle \\ \dot{\mathbf{X}} &= \dot{k} \cdot (\mathbf{X}_j - \mathbf{X}_i)\end{aligned}\tag{25}$$

Průběh parametru $k(t)$ v čase definuje profil pohybu.

2. V každém diskretním časovém okamžiku $t = t_k, T = t_{k+1} - t_k$, kde T je perioda vzorkování, je spuštěna diskretní podoba algoritmu (24):

$$\mathbf{Q}_k^{l+1} = \mathbf{Q}_k^l + \mathbf{J}^\dagger(\mathbf{Q}_k^l) \cdot \left(\dot{\mathbf{X}}_{d_k} + \begin{bmatrix} k_p \cdot \mathbf{I} & \mathbf{0} \\ \mathbf{0} & k_o \cdot \mathbf{I} \end{bmatrix} \begin{bmatrix} e_{p_k}^l \\ e_{o_k}^l \end{bmatrix} \right) + \left(\mathbf{I} - \mathbf{J}^\dagger(\mathbf{Q}_k^l) \cdot \mathbf{J}(\mathbf{Q}_k^l) \right) \cdot \frac{\partial w(\mathbf{Q}_k^l)}{\partial \mathbf{Q}_k^l}\tag{26}$$

kde počáteční podmínku $\mathbf{Q}_k^0 = \mathbf{Q}_{k-1}$ tvoří známé polohy kloubových souřadnic \mathbf{Q}_k z předchozího časového okamžiku. Algoritmus iteruje v l dokud není chyba (v translaci e_p resp. orientaci e_o pod daným limitem).

◆

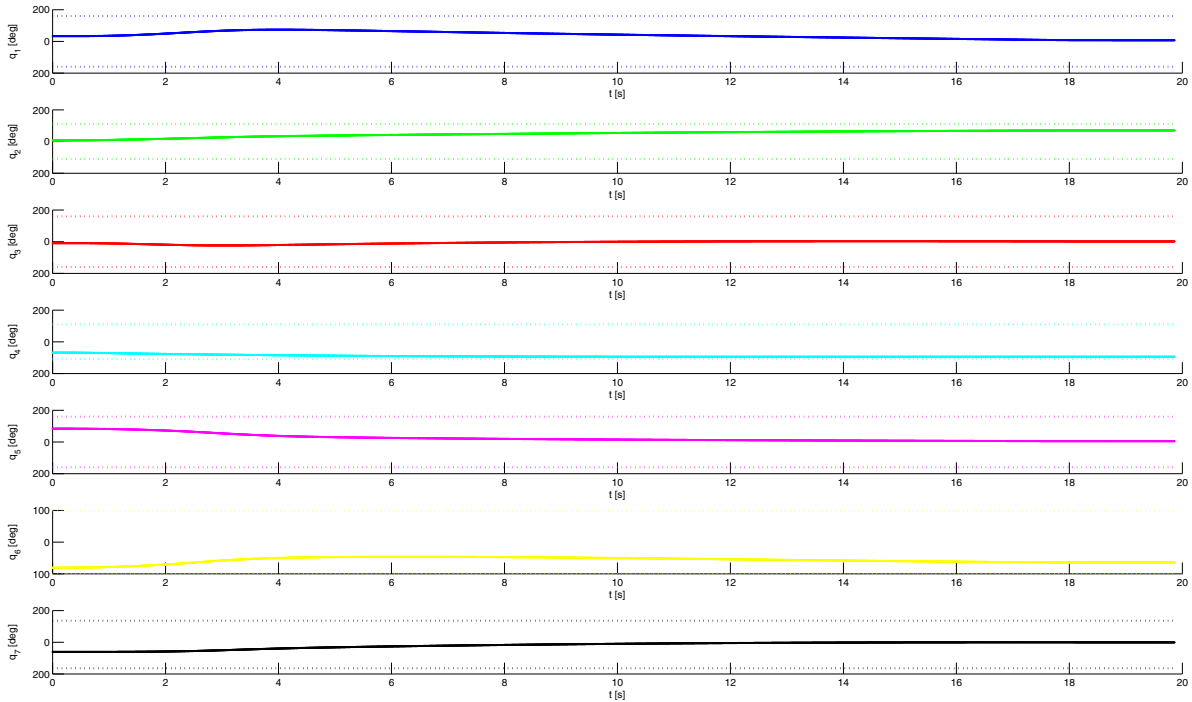
Poznámky:

- Konstanty k_p, k_o jsou navíc dynamicky měněny v průběhu podél trajektorie (s rostoucím k) od malých do větších hodnot z důvodu generování přiměřených kloubových rychlostí na počátku přímkové trajektorie na úkor polohových chyb.
- Konstanta k_o v kritériu $w(\mathbf{Q})$ ovlivňuje rychlost konvergence k optimálnímu řešení.
- Přesto, že počáteční \mathbf{Q}_i i koncový bod \mathbf{Q}_j přímkové trajektorie byl vygenerován optimálním JOGem se shodným kritériem optimality a lze tudíž předpokládat, že numerický algoritmus podél přímkové trajektorie by měl dokonvergovat na konci přímkového úseku do hodnoty \mathbf{Q}_j (identické konfigurace robotu jako při procesu učení), není tento předpoklad vlivem numerických chyb v iteracích algoritmu striktně splněn (v některých případech, kdy přímkový úsek probíhá v těsné blízkosti singularit může dokonce dojít k bifurkaci řešení a cílová konfigurace bude zcela odlišná byť s koncovým efektem na správném místě - teoreticky). Vzhledem k výše uvedenému požadavku, že robot musí na konci přímkového úseku skončit přesně v naučených kloubových polohách \mathbf{Q}_j (v důsledku nepřesností v kinematické kalibraci robotu), je na konci každého kroku numerického algoritmu realizován ještě kloubový (nekoordinovaný) pohyb z výsledných kloubových poloh \mathbf{Q}_j^l do naučených poloh \mathbf{Q}_j . Rozdíl $\|\mathbf{Q}_j^l - \mathbf{Q}_j\|$ je díky nepřesnosti numerických výpočtů však natolik nevýznamný, že při takové korekci není pohyb koncového efektoru v podstatě patrný a nijak nevádí cílové aplikaci. Uvedená korekce je zároveň nezbytná v případě navazování přímkových úseků trajektorií
- Uvedeným způsobem jsou generovány všechny kombinace přímkových úseků, viz Obrázek 3, a uloženy jako předimplemetované pohyby kloubů manipulátoru.
Např.: Přímkový úsek z „Approach“ polohy Analyzátoru do Domovské polohy „Home“:
 $\mathbf{Q}_{45} \rightarrow$ přímkový pohyb $\rightarrow \mathbf{Q}_{30}$

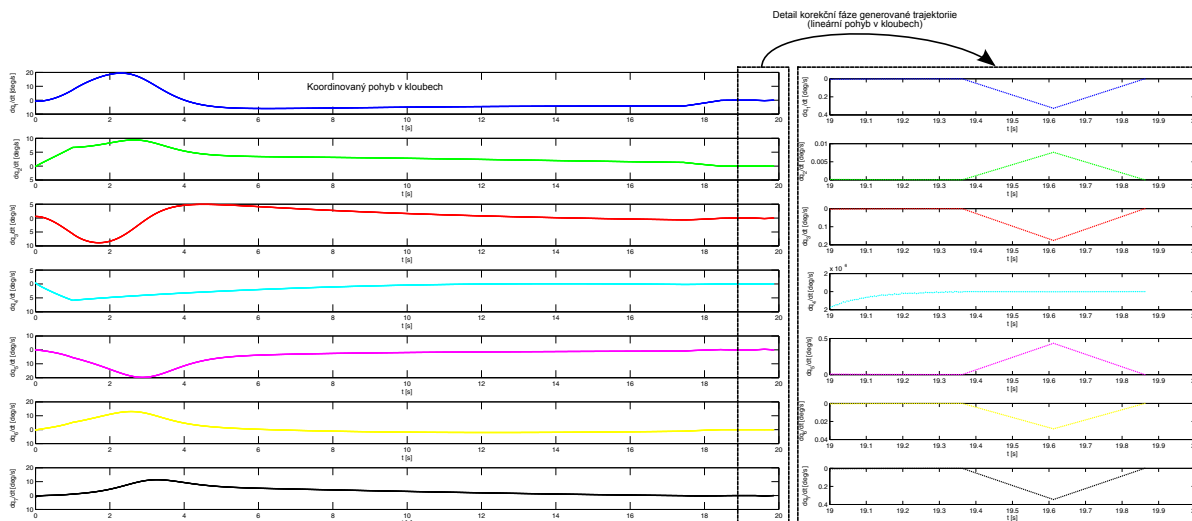
⁶Poznamenejme, že interpolace v orientaci byla realizována přes Eulerovy úhly (minimální reprezentace) a následně přepočtena na kvaterniony.

- Komplexní trajektorie pohybu jsou dále složeny z dílčích přímkových úseků.
Např.: Založení z domovské Domovské polohy „Home“ do 1. pozice v Pipetovačce: $\{Q_{30} \rightarrow \text{přímkový pohyb} \rightarrow Q_2\} + \{Q_2 \rightarrow \text{přímkový pohyb (s jiným profilem rychlosti)} \rightarrow Q_1\}$, atd.

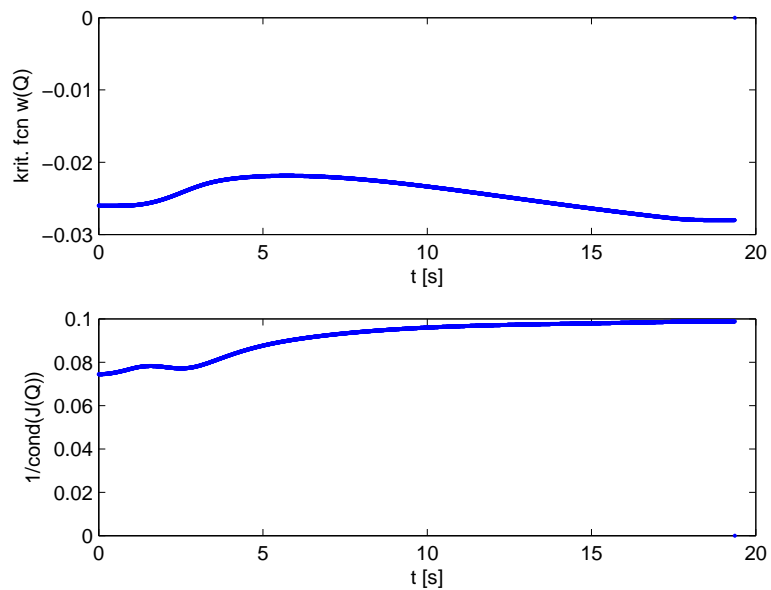
Následující obrázky znázorní jednu realizaci vypočtených kloubových souřadnic pro přímkový pohyb z Q_{42} do Q_{30} .



Obrázek 4: Generované polohy kloubových souřadnic s vyznačenými maximálními povolenými limity aktuátorů.



Obrázek 5: Generované rychlosti kloubových souřadnic s detailem zmíněné korekční fáze.



Obrázek 6: Hodnoty kritériální funkce a převrácená hodnota čísla podmíněnosti kinematického jakobiánu manipulátoru (míra vzdálenosti od singulárních poloh).

4 Závěr

Předložená technická zpráva shrnuje výzkumné výsledky, které byly využity v implementaci řídicího systému manipulátoru SCHUNK LWA 4D a jeho nasazení v pilotní aplikaci, viz [6]. Výsledky lze shrnout následovně:

- **Virtuální simulační model**

Byl vytvořen virtuální simulační model manipulátoru v programovém prostředí Matlab/Simulink/SimMechanics za účelem simulačního ověření pokročilých metod řízení pohybu (optimální koordinovaný JOG, simulace generovaných trajektorií pro pick and place aplikace).

- **Algoritmus pro výpočet kinematických úloh**

Byl odvozen matematický popis dopředné i zpětné kinematiky manipulátoru. Vzhledem k faktu, že navržený manipulátor je redundantní (7 DoF), byla inverzní kinematická úloha parametrizována redundantní kloubovou souřadnicí (polohou dodatečného kloubu), která nemá paralelu v kinematických architekturách standardních 6 DoF průmyslových manipulátorů.

- **Optimální koordinovaný JOGovací algoritmus**

Přesto, že navržený matematický model pro inverzní kinematickou úlohu redundantního manipulátoru umožňuje kontinuálně měnit vnitřní konfiguraci manipulátoru (uspořádání ramen) pro předepsanou trajektorii koncového efektoru prostřednictvím redundantní kloubové souřadnice, a tím umožňuje operátorovi manuálně optimalizovat pohyb manipulátoru, je takový postup naprosto neintuitivní a v praxi v podstatě nepoužitelný. Právě proto byl vyvinut algoritmus pro optimální koordinovaný JOGovací režim, ve kterém operátor pouze diktuje základních 6 DoF koncového efektoru (konkrétně rychlost translace a rotace) a rychlosti kloubů manipulátoru jsou dopočítávány takovým způsobem, aby bylo minimalizováno definované kritérium. Takovým kritériem byl zvolen kompromis mezi udržením poloh kloubových souřadnic uprostřed povolených limitů aktuátorů a minimalizace rychlostí aktuátorů. Algoritmus byl doplněn o bezpečnostní subsystém limitující maximální kloubové rychlosti na úkor zpomalení koordinovaného pohybu koncového efektoru.

- **Generátor trajektorie koncového efektoru**

Za účelem plánování pohybu manipulátoru v pilotní aplikaci byl realizován algoritmus generování kloubových souřadnic manipulátoru pro přímkové přejezdy koncového efektoru manipulátoru mezi naučenými body v pracovním prostoru (body naučeny operátorem za pomoci optimálního JOGovacího režimu). Generátor trajektorie, jehož výstupem byly uložené datové soubory kloubových souřadnic pro dílčí předgenerované trajektorie využíval identické kritérium optimality jako JOGovací režim.

Poděkování

Tento výzkum byl podpořen z projektu č. FV 10044 v programu TRIO Ministerstva průmyslu a obchodu.

Reference

- [1] Denavit, J.; Hartenberg, R. S.: A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, ročník 22, 1955: s. 215–221.
URL <http://ci.nii.ac.jp/naid/10008019314/en/>
- [2] Khalil, W.; Dombre, E.: *Modeling, Identification and Control of Robots*. Kogan Page Science paper edition, Elsevier Science, 2004, ISBN 9780080536613.
URL <http://books.google.cz/books?id=nyrY0Pu5k10C>
- [3] Sciavicco, L.; Siciliano, B.: *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing, Springer London, 2000, ISBN 9781852332211.
URL <http://books.google.fr/books?id=v9PLbcYd9aUC>
- [4] Siciliano, B.: Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, ročník 3, č. 3, 1990: s. 201–212, ISSN 0921-0296, doi:10.1007/BF00126069.
URL <http://dx.doi.org/10.1007/BF00126069>
- [5] The MathWorks, I.: Matlab (Simulink, SimMechanics).
URL <http://www.mathworks.com/>
- [6] Čechura, T.; Jäger, A.; Švejda, M.: Technická a uživatelská dokumentace k softwaru „Řídicí systém ramene SCHUNK LWA 4D“. Technická zpráva, Západočeská univerzita v Plzni, 2017.
- [7] Švejda, M.: Kinematika robotických architektur. Katedra Kybernetiky, ZČU v Plzni (online: http://home.zcu.cz/~msvejda/_publications/2011/rigo.pdf), 2011.
- [8] Švejda, M.: Úvod do robotiky a mechatroniky. Přednášky k předmětu, 2012.
URL <http://home.zcu.cz/~msvejda/URM/>
- [9] Švejda, M.: *Optimalizace robotických architektur*. Dizertační práce, Západočeská univerzita v Plzni, 2016.
URL http://home.zcu.cz/~msvejda/_publications/2016/4_SvejdaMartin_thesis_2016_06_14.pdf

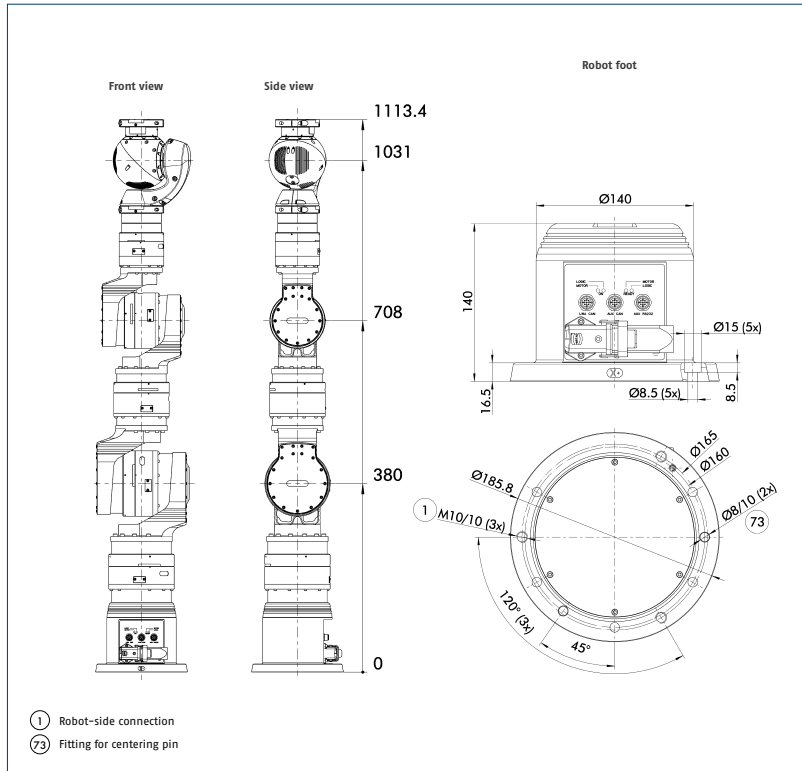
A Parametry manipulátoru SCHUNK LWA 4D

A.1 7 DoF robot SCHUNK LWA 4D

LWA 4D

Dextrous Lightweight Arm

Technical data



Designation	LWA 4D	Power supply	24 V DC / avg. 5 A / max. 15 A
ID	0306962	Interface	CANopen (GIA DS402:IEC61800-7-201)
Type	7 DOF lightweight robot	Axes	Speed with nominal load Range
Number of axes	7	Axis 1	40°/s ±180°
Max. payload [kg]	10	Axis 2	40°/s ±123°
Repeat accuracy [mm]	±0.15	Axis 3	40°/s ±180°
Position feedback	Pseudo-absolute position measuring	Axis 4	40°/s ±125°
Drives	Brushless servomotors with permanent-magnet brake	Axis 5	40°/s ±180°
Pan-tilt unit flange	Flat tool changer with free lines and power supply	Axis 6	72°/s ±170°
Installation direction	Any	Axis 7	72°/s ±170°
Dead weight [kg]	18	Grippers	WSG 50, PG-plus 70, MEG, SDH 2, SVH
IP class [IP]	54	Changer	FWS 115
		Robot control system	R05 node (ros.org)

23

SCHUNK

A.2 Force/Torque jednotka FTM 45

Datasheet obsahuje informace k jednotce FTM 75, která se od FTM 45 liší pouze rozsahem měření síly/momentu.

FTM 75

Force-Torque Module

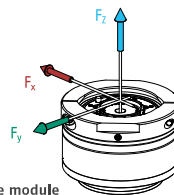
Compact and flat. With integrated Measuring and Evaluation Technology.



The FTM 75 and FTM 115 force-torque modules are measuring systems for all six degrees of freedom, and thus make it possible to control and regulate the process forces in the robot. In the event of a collision, it is also possible for a lightweight arm to "give" flexibly and for compensation scenarios to be introduced.

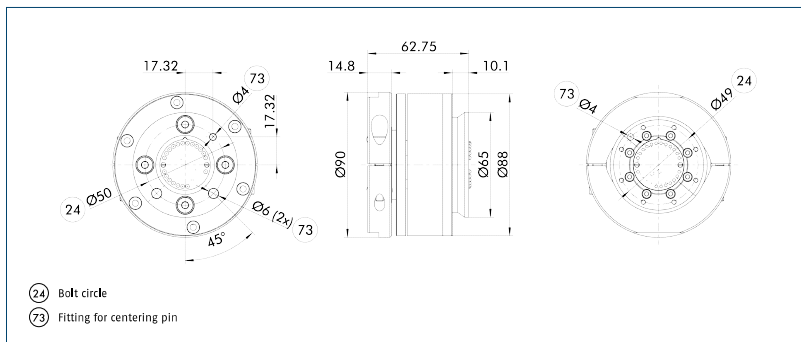
Your advantage:

- Compact and flat design



FTM 75 force-torque module

Technical data



Designation	FTM 75 force-torque module	Resolution M_x, M_y, M_z [Nm]:	0.01
ID	31001023	Dead weight [kg]:	0.7
Range of measurement F_x, F_y [N]:	±300	Power supply [V DC]:	24
Range of measurement F_z [N]:	±300	Fieldbus interface	CAN
Range of measurement M_x, M_y [Nm]:	±15		
Range of measurement M_z [Nm]:	±15		
Resolution F_x, F_y, F_z [N]:	0.3		

A.3 Uchopovač PG+ 70

PG-plus 70

Servo-electric 2-Finger Parallel Gripper with FWA 115 Flat Change System

Universally mobile Applicability. Sensitive Gripping Force Regulation for a large Variety of Parts.



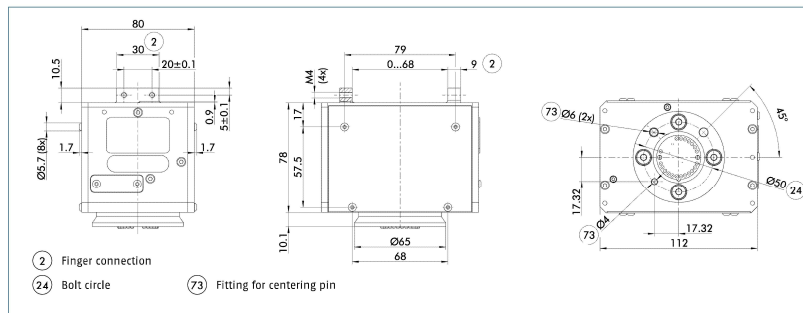
PG-plus 70
2-finger parallel
grripper
In application
on page 15

The SCHUNK PG-plus 70 is a servo-electric 2-finger parallel gripper with sensitive grip force control and long stroke. It is universally applicable, highly flexible with a large variety of parts, and suitable for sensitive components in a clean working environment. The control, regulating and power electronics are completely integrated. The PG-plus 70 is also available in a dust-proof version.

Your advantage:

- **Gripping force configuration from 30 – 200 N** available for fine gripping of sensitive workpieces
- **Long stroke of 60 mm** for flexible workpiece handling

Technical data



Designation	PG-plus 70	Dust-proof version	Max. acceleration [mm/s ²]	328
ID	31000796	31000860	Dead weight [kg]	1.2
Stroke per finger [mm]	30		IP class [IP]	20
Min. / max. Gripping force [N]	30 / 200		Min. / max. Ambient temperature [°C]	5 / 55
Rated force [N]	160		Power supply [V DC]	24
Recommended workpiece weight [kg]	1.0		Rated current [A]	1.4
Max. permissible finger length (at rated force) [mm]	140		Max. total current [A]	1.8
Repeat accuracy [mm]	±0.05		Parameterization interface	CAN / USB
Max. speed [mm/s]	82		Fieldbus interface	CANopen DS 402

B Inverzní kinematika (polohové závislosti - Matlab)

```

1 O50 = O70 - R70(:,3)*L4;
2
3 % theta4
4 C4 = -(L2^2+L3^2-O50(1)^2-O50(2)^2-(O50(3)-L1)^2)/(2*L2*L3);
5 if sol(2) == 1
6     S4 = +realsqrt(1-C4^2);
7 else
8     S4 = -realsqrt(1-C4^2);
9 end
10 theta4 = atan2(S4,C4);
11
12 % theta1
13 S3 = sin(theta3);
14 C3 = cos(theta3);
15 if sol(1) == 1
16     if abs(O50(1))<1e-6
17         C1 = (-L3*S4*S3)/O50(2);
18         S1 = +sqrt(1-C1^2);
19     else
20         if O50(1) > 0
21             S1 = (O50(2)*realsqrt(O50(1)^2*(O50(2)^2+O50(1)^2- ...
22                 L3^2*S4^2*S3^2)) + L3*S4*S3*O50(1)^2) ...
23                 /((O50(2)^2+O50(1)^2)*O50(1));
24             C1 = (-O50(2)*L3*S4*S3+realsqrt(O50(1)^2*(O50(2)^2 ...
25                 +O50(1)^2-L3^2*S4^2*S3^2)))/(O50(2)^2+O50(1)^2);
26         else
27             S1 = (-O50(2)*realsqrt(O50(1)^2*(O50(2)^2+ ...
28                 O50(1)^2-L3^2*S4^2*S3^2)) ...
29                 +L3*S4*S3*O50(1)^2)/((O50(2)^2+O50(1)^2)*O50(1));
30             C1 = (-O50(2)*L3*S4*S3-realsqrt(O50(1)^2*(O50(2)^2 ...
31                 +O50(1)^2-L3^2*S4^2*S3^2)))/(O50(2)^2+O50(1)^2);
32         end
33     end
34 else
35     if abs(O50(1))<1e-6
36         C1 = (-L3*S4*S3)/O50(2);
37         S1 = -sqrt(1-C1^2);
38     else
39         if O50(1) > 0
40             S1 = (-O50(2)*realsqrt(O50(1)^2*(O50(2)^2 + ...
41                 O50(1)^2-L3^2*S4^2*S3^2)) + ...
42                 L3*S4*S3*O50(1)^2)/((O50(2)^2+O50(1)^2)*O50(1));
43             C1 = (-O50(2)*L3*S4*S3-realsqrt(O50(1)^2*(O50(2)^2 + ...
44                 O50(1)^2-L3^2*S4^2*S3^2)))/(O50(2)^2+O50(1)^2);
45         else
46             S1 = (O50(2)*realsqrt(O50(1)^2*(O50(2)^2 + ...
47                 O50(1)^2-L3^2*S4^2*S3^2)) + L3*S4*S3*O50(1)^2) / ...
48                 ((O50(2)^2+O50(1)^2)*O50(1));
49             C1 = (-O50(2)*L3*S4*S3+realsqrt(O50(1)^2*(O50(2)^2 + O50(1)^2 - ...
50                 L3^2*S4^2*S3^2)))/(O50(2)^2+O50(1)^2);
51         end
52     end
53 end
54 theta1 = atan2(S1,C1);
55
56 % theta2
57 if abs(S1) > abs(C1)
58     S2 = -0.1e1 / S1 / (L3 ^ 2 * S4 ^ 2 * C3 ^ 2 + L3 ^ 2 * C4 ^ 2 + 0.2e1 * ...
59         L3 * C4 * L2 + L2 ^ 2) * (O50(2) * L3 * C4 + O50(2) * L2 + L3 ^ 2 * ...
60         S4 * C1 * S3 * C4 + L3 * S4 * C1 * S3 * L2 + L3 * S4 * S1 * C3 * ...
61         O50(3) - L3 * S4 * S1 * C3 * L1);
62     C2 = -(C4 * L3 * S1 * L1 - C4 * L3 * S1 * O50(3) + C3 * S4 ^ 2 * C1 * S3 ...
63         * L3 ^ 2 + C3 * S4 * O50(2) * L3 + S1 * L1 * L2 - S1 * O50(3) * L2) / ...

```

```

        S1 / (L3 ^ 2 * S4 ^ 2 * C3 ^ 2 + L3 ^ 2 * C4 ^ 2 + 0.2e1 * L3 * C4 * ...
        L2 + L2 ^ 2);
48 else
49     S2 = -0.1e1 / C1 / (L3 ^ 2 * S4 ^ 2 * C3 ^ 2 + L3 ^ 2 * C4 ^ 2 + 0.2e1 * ...
        L3 * C4 * L2 + L2 ^ 2) * (O50(1) * L3 * C4 + O50(1) * L2 - L3 ^ 2 * ...
        S4 * S1 * S3 * C4 - L3 * S4 * S1 * S3 * L2 + C1 * L3 * S4 * C3 * ...
        O50(3) - C1 * L3 * S4 * C3 * L1);
50     C2 = (C4 * C1 * L3 * O50(3) - C4 * C1 * L3 * L1 + C3 * S4 ^ 2 * S1 * S3 * ...
        L3 ^ 2 - C3 * S4 * O50(1) * L3 - C1 * L1 * L2 + C1 * O50(3) * L2) / ...
        C1 / (L3 ^ 2 * S4 ^ 2 * C3 ^ 2 + L3 ^ 2 * C4 ^ 2 + 0.2e1 * L3 * C4 * ...
        L2 + L2 ^ 2);
51 end
52 theta2 = atan2(S2,C2);
53
54 % theta5,6,7
55 T10 = DH([L1,theta1,0, pi/2]);
56 T21 = DH([0,theta2,0,-pi/2]);
57 T32 = DH([L2,theta3,0,pi/2]);
58 T43 = DH([0,theta4,0, -pi/2]);
59
60 T40 = T10*T21*T32*T43;
61 R40 = T40(1:3,1:3);
62
63 R74 = R40'*R70;
64
65 C6 = R74(3,3);
66 if sol(3) == 1
67     S6 = +realsqrt(R74(3,1)^2 + R74(3,2)^2);
68 else
69     S6 = -realsqrt(R74(3,1)^2 + R74(3,2)^2);
70 end
71
72 S7 = -R74(3,2)/S6;
73 C7 = R74(3,1)/S6;
74 S5 = -R74(2,3)/S6;
75 C5 = -R74(1,3)/S6;
76
77 theta6 = atan2(S6,C6);
78 theta7 = atan2(S7,C7);
79 theta5 = atan2(S5,C5);
80
81 Q = [theta1;theta2;theta3;theta4;theta5;theta6;theta7];

```