TAČR: Advanced Robotic Architectures for Industrial Inspection (ADRA-2I)

Project Number: TF02000041

Preliminary architecture design and control algorithm of ROBIN robot

(Research report)

Martin Švejda (ZCU)

20 January 2017

Company ID:
ZCU - University of West Bohemia, SM - SmartMotion s.r.o., UJV - ÚJV Řež, a. s

Abstract

The technical report reveals the possibilities of kinematic control of the first part of a new snake-like robot architecture ROBIN (ROBotic INspection). The first part of the proposed robot consists of circumferential travel around the pipe and three 2 DoF cells with two perpendicular rotation axes. The proposed control algorithm makes possible to control the position and (reduced) orientation of the end-effector (3 translation and 1 rotation DoFs which are sufficient for positioning/orientating of a testing probe with an expected surface contact) as well as the distances of robot elbows from pipe surface (robot body shape control for obstacle avoidance or minimizing of occupied space).

Contents

1 Introduction

ROBIN (ROBotic INspection) is new robotic conceptual design of hybrid robotic architecture for Non Destructive Testing (NDT) of the pipe weld joints of complex geometries. The report deals with the preliminary kinematic architecture design and design of intuitive control algorithms for the first part (conventional part) of the ROBIN robot. The ROBIN is supposed to be snake-like robot where the first part of the robot consists of three 2 DoF "snake cells" witch revolute actuators and circumeferential travel around the pipe forming the base of the manipulator. Note that the system of circumferential travel was inspired by previous TAČR projects *Research, development and evaluation of a new technology for modern weld diagnostics in nuclear power plants* No. TA01020457, [9, 10, 4] and *Centre for Advanced Nuclear Technologies (CANUT)* No. TE01020455, [6]. The second part of the robot is supposed to be designed as special architecture cells based on Nitinol Wire (Muscle Wire) actuators. The second part of the robot is beyond the scope of this report and the relevant information and basic kinematic/dynamic modelling of the nitinol wire actuators cane be found in [5].

The CAD concept of the conventional part of the ROBIN is shown in Figure 1. The standard Denavit-Hartenberg (D-H) notation [1] of the manipulator with assigned coordinate systems (CSs) is depicted in Figure 2 and Denavit-Hartenberg kinematic parameters are listed in Table 1.

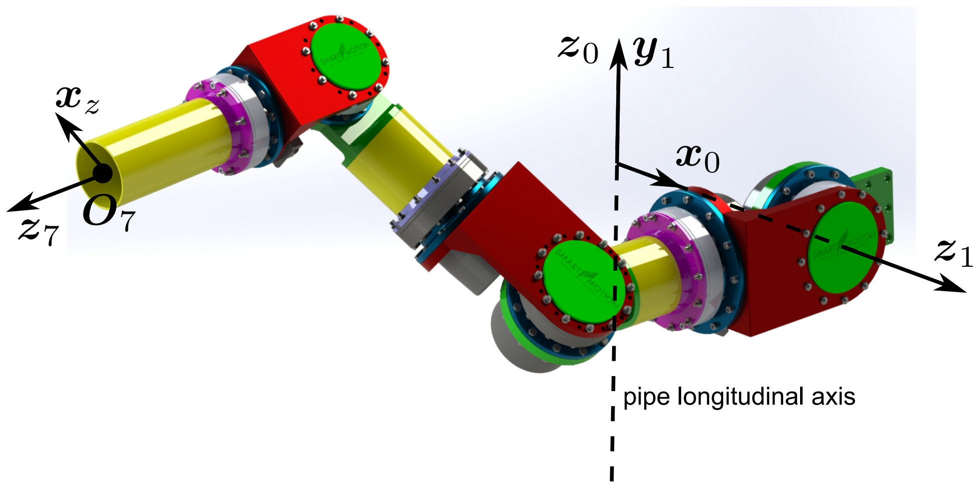

Figure 1: CAD layout of the first (conventional part) of the ROBIN robot (without circumferential travel)

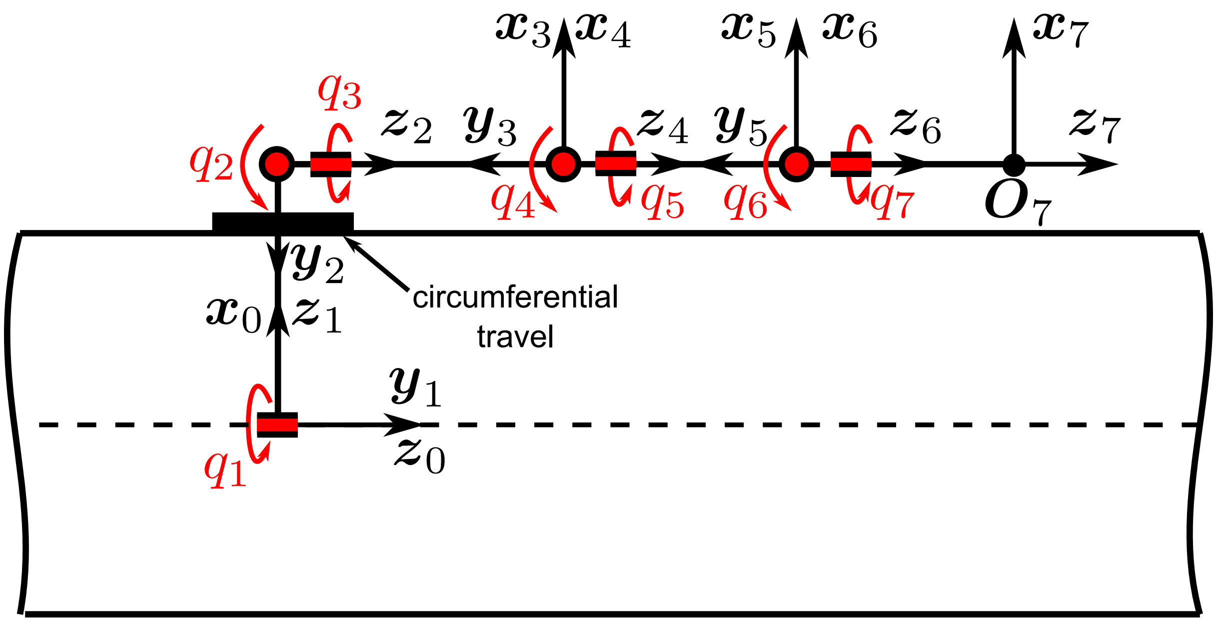

Figure 2: Denavit-Hartenberg notation for ROBIN robot.

i	d_i	θ_i	a_i	α_i
1	0	$theta_1$	0	$\frac{\pi}{2}$
2	L_1	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

Table 1: Denavit-Hartenberg parameters

The position and orientation of the end-effector are formulated via generalized coordinates[1]:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{O}_7^0 & \boldsymbol{R}_7^0 \end{bmatrix}, \quad \boldsymbol{R}_7^0 = \begin{bmatrix} \boldsymbol{x}_7^0 & \boldsymbol{y}_7^0 & \boldsymbol{z}_7^0 \end{bmatrix}$$
$$\text{or (alternatively)} \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{O}_7^0 & \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \end{bmatrix}, \quad \alpha,\ \beta,\ \gamma \ \text{ are XYZ Euler angles.} \tag{1}$$

where $\boldsymbol{O}_7^0$ are position coordinates and $\boldsymbol{R}_7^0$ is rotation matrix of the end-effector with respect to CS F_0. The representation of the end-effector orientation can be reformulated as XYZ Euler angles α, β, γ (rotation around $\boldsymbol{x}$ axis by the angle α, rotation around a new $\boldsymbol{y}$ axis by the angle β and rotation around a new $\boldsymbol{z}$ axis by the angle γ). The direct and inverse transformation between Euler Angles and rotation matrix (including representation singularities for inverse problem, $\boldsymbol{R}_7^0 \Rightarrow [\alpha, \beta, \gamma]$) can be found in [8, 2].

Joint coordinates are represented by the position of the circumferential travel $\theta_1 [rad]$ and position of the revolute actuators $\theta_2 \ldots \theta_7$ [rad]:

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & a_2 & \ldots & q_7 \end{bmatrix}^T = \begin{bmatrix} \theta_1 & \theta_2 & \ldots \theta_7 \end{bmatrix}^T \tag{2}$$

[1] $\boldsymbol{X}_i^j$ denotes the point/vector/matrix $\boldsymbol{X}_i$ with respect to CS F_j.

2 Standard coordinate control of ROBIN robot

Standard kinematic computations are *direct kinematic* and *inverse kinematic model.* **Direct geometric model (DGM)** ($\boldsymbol{Q} \Rightarrow \boldsymbol{X}$) arises from kinematic description using homogeneous transformation matrices (dependent on D-H parameters and joint coordinates) and it can be formulated as follows:

$$
\begin{aligned}
&\boldsymbol{T}_7^0 = \prod_{i=1}^{7} \boldsymbol{T}_i^{i-1}(d_i, \theta_i, a_i, \alpha_i), \quad \boldsymbol{T}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\boldsymbol{O}_7^0 = \boldsymbol{T}_7^0[1:3,4], \quad \boldsymbol{R}_7^0 = \boldsymbol{T}_7^0[1:3,1:3] \;\Rightarrow\; \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T
\end{aligned} \tag{3}
$$

where D-H parameters are summarized in Table 1 and joint coordinates are represented as θ_i D-H parameters (position of the actuators $\boldsymbol{Q} = [q_i] = [\theta_i]$, $i = 1, \ldots, 7$).

Inverse geometric model (IGM) ($\boldsymbol{X} \Rightarrow \boldsymbol{Q}$) is given by a non-trivial solution which can be formulated as Algorithm 1. Because there are more independent joint coordinates (7) than independent generalized coordinates (3 translation and 3 rotation) the robot under consideration is redundant and therefore one joint coordinate can be understood as parameter which makes possible to parametrized all (infinity) solution of the IGM.

♦ Algorithm 1 (Standard IGM for ROBIN robot)

Algorithm input:

Generalized coordinates (position and orientation of the end-effector):

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{O}_7^0 & \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \end{bmatrix}$$

Redundant joint coordinate (position of circumferential travel):

$$\boldsymbol{Q}_{red} = q_1$$

Solution of IGM (8 isolated solutions):

$$sol = \begin{bmatrix} i & j & k \end{bmatrix}, \quad i,\, j,\, k = \{1, 2\}$$

Algorithm outputs:

Joint coordinates (only positions are considered):

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{bmatrix}^T$$

Algorithm parameters:

Length of the links:

$$\boldsymbol{\xi} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T$$

Layout of the algorithm (without implementation details):

1. Euler angles to rotation matrix computation (for details see [8, 7]):

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \rightarrow \boldsymbol{R}_7^0 = \begin{bmatrix} \boldsymbol{x}_7^0 & \boldsymbol{y}_7^0 & \boldsymbol{z}_7^0 \end{bmatrix} \tag{4}$$

2. The center of the spherical wrist $\boldsymbol{O}_5$ is given as[2]:

$$\begin{aligned} \boldsymbol{O}_5^0 &= \boldsymbol{O}_7^0 - L_4 \boldsymbol{z}_7^0 \quad \text{(with respect to CS } F_0\text{)} \\ \boldsymbol{O}_5^1 &= \begin{bmatrix} S_1 \boldsymbol{O}_5^0[2] + C_1 \boldsymbol{O}_5^0[1] \\ \boldsymbol{O}_5^0[3] \\ S_1 \boldsymbol{O}_5^0[1] - C_1 \boldsymbol{O}_5^0[2] \end{bmatrix} \quad \text{(with respect to CS } F_1\text{)} \end{aligned} \tag{5}$$

3. Computation of q_4 coordinate:

$$\begin{aligned} C_4 &= \frac{1}{2} \frac{\boldsymbol{O}_5^1[1]^2 + \boldsymbol{O}_5^1[2]^2 + \boldsymbol{O}_5^1[3]^2 + L_1^2 - L_2^2 - L_3^2 - 2L_1 \boldsymbol{O}_5^1[3]}{L_3 L_2} \\ S_4 &= \begin{cases} +\sqrt{1 - C_4^2} & \text{for} \quad sol = \begin{bmatrix} i & 1 & k \end{bmatrix} \\ -\sqrt{1 - C_4^2} & \text{for} \quad sol = \begin{bmatrix} i & 2 & k \end{bmatrix} \end{cases} \\ q_4 &= \theta_4 = \operatorname{atan2}(S_4, C_4) \end{aligned} \tag{6}$$

4. Computation of q_3 coordinate:

$$\begin{aligned} S_3 &= -\frac{\boldsymbol{O}_5^1[3] - L_1}{L_3 S_4} \\ C_3 &= \begin{cases} +\sqrt{1 - S_3^2} & \text{for} \quad sol = \begin{bmatrix} 1 & j & k \end{bmatrix} \\ -\sqrt{1 - S_3^2} & \text{for} \quad sol = \begin{bmatrix} 2 & j & k \end{bmatrix} \end{cases} \\ q_3 &= \theta_3 = \operatorname{atan2}(S_3, C_3) \end{aligned} \tag{7}$$

5. Computation of q_2 coordinate:

$$\begin{aligned} S_2 &= \frac{-L_3 \boldsymbol{O}_5^1[1] C_4 + C_3 S_4 \boldsymbol{O}_5^1[2] L_3 - \boldsymbol{O}_5^1[1] L_2}{L_3^2 C_4^2 + 2 L_3 L_2 C_4 + L_2^2 + L_3^2 C_3^2 S_4^2} \\ C_2 &= \frac{L_3 \boldsymbol{O}_5^1[2] C_4 + C_3 S_4 \boldsymbol{O}_5^1[1] L_3 + \boldsymbol{O}_5^1[2] L_2}{L_3^2 C_4^2 + 2 L_3 L_2 C_4 + L_2^2 + L_3^2 C_3^2 S_4^2} \\ q_2 &= \theta_2 = \operatorname{atan2}(S_2, C_2) \end{aligned} \tag{8}$$

6. Computation of q_5, q_6, q_7 coordinates (spherical wrist):

   The homogeneous transformation matrix $\boldsymbol{T}_4^0$ (position and orientation) of CS F_4 are given by known joint coordinates $q_1, \ldots, q_4$. Therefore the relative transformation between CS F_4 and CS F_7 dependent on unknown joint coordinates q_5, q_6, q_7 is:

$$\boldsymbol{R}_7^4(q_5, q_6, q_7) = \begin{bmatrix} \star & \star & C_5 S_6 \\ \star & \star & S_5 S_6 \\ -S_6 C_7 & S_6 S_7 & C_6 \end{bmatrix} \stackrel{!}{=} \underbrace{\left(\boldsymbol{R}_4^0\right)^T \boldsymbol{R}_7^0}_{\text{known part}} \tag{9}$$

[2] S_i resp. C_i denote $\sin(\theta_i)$ resp. $\cos(\theta_i)$. $\boldsymbol{X}[i, j]$ denotes element of vector/matrix $\boldsymbol{X}$, $\boldsymbol{X}[i : k, k : l]$ denote subvector/submatrix of $\boldsymbol{X}$ (see Matlab notation).

It results in:

$$C_6 = \boldsymbol{R}_7^4[3,3], \quad S_6 = \begin{cases} +\sqrt{1-C_6^2} & \text{for} \quad sol = \begin{bmatrix} i & j & 1 \end{bmatrix} \\ -\sqrt{1-C_6^2} & \text{for} \quad sol = \begin{bmatrix} i & j & 2 \end{bmatrix} \end{cases}$$
$$q_6 = \theta_6 = \text{atan2}(S_6, C_6) \tag{10}$$

$$S_5 = \frac{\boldsymbol{R}_7^4[2,3]}{S_6}, \quad C_5 = \frac{\boldsymbol{R}_7^4[1,3]}{S_6}, \quad q_5 = \theta_5 = \text{atan2}(S_5, C_5) \tag{11}$$

$$S_7 = \frac{\boldsymbol{R}_7^4[3,2]}{S_6}, \quad C_7 = -\frac{\boldsymbol{R}_7^4[3,1]}{S_6}, \quad q_7 = \theta_7 = \text{atan2}(S_7, C_7) \tag{12}$$

♦

3 Advanced algorithm for coordinate control of ROBIN robot

The standard control algorithm, see Algorithm 1, is convenient for standard applications where the full position and orientation control are required. Moreover, in that cases the circumferential travel brings some ambiguities and the next demand has to be specified for uniquely IGM solution computation. But for NDT applications of pipe welds only the $\boldsymbol{xyz}$ position (translation) of the end-effector has to be controlled because the NDT probe tilting movement (2 rotation DoFs) and head orientation (1 translation DoF) can be resolved as 2 DoF passive element (given by the pipe surface contact) with 1 rotation DoF simple servo for probe head orientating, see Figure 3.

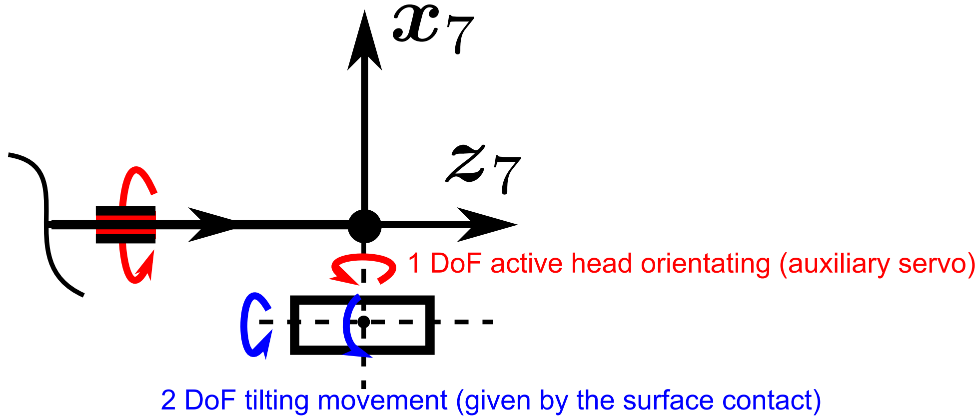

Figure 3: Orientating of NDT probe (only idea without technical details)

Therefore it is necessary to find modified control algorithm for IGM solution. Considering the previous experiences with robotic NDT applications a new idea for IGM was introduced. The generalized coordinates were modified in order to intuitive control of ROBIN robot with following demands on control generalized coordinates:

- 3 DoF: $\boldsymbol{O}_7^0 = \begin{bmatrix} x & y & z \end{bmatrix}^T \Rightarrow$ (Translation) control of the end-effector - positioning of the NDT probe (orientation is given by the passive element and auxiliary servo - not addressed in this technical report)
- 2 DoF: r_3, r_5 $\Rightarrow$ Distance control of elbows (origin $\boldsymbol{O}_3$, $\boldsymbol{O}_5$ of CSs F_3, F_5) from pipe longitudinal axis (equivalently from pipe surface) - measure of distances of "snake-like" robot body from pipe surface (to ensure encircling the pipe)

- 1 DoF: $\alpha \Rightarrow$ The last robot link orientation given by the angle around normal to pipe surface (an angle between the last link projection into tangent plane of pipe surface at $\boldsymbol{O}_7^0$ and perpendicular to the longitudinal pipe axis) - robot body shaping for obstacle avoidance
- 1 DoF (remaining): The last joint coordinate is neglected (set to $\theta_7 = 0$) and it is not used for control (it will be used for control of the second part (nitinol actuators) of robot)

New generalized coordinates meaning is depicted in Figure 4.

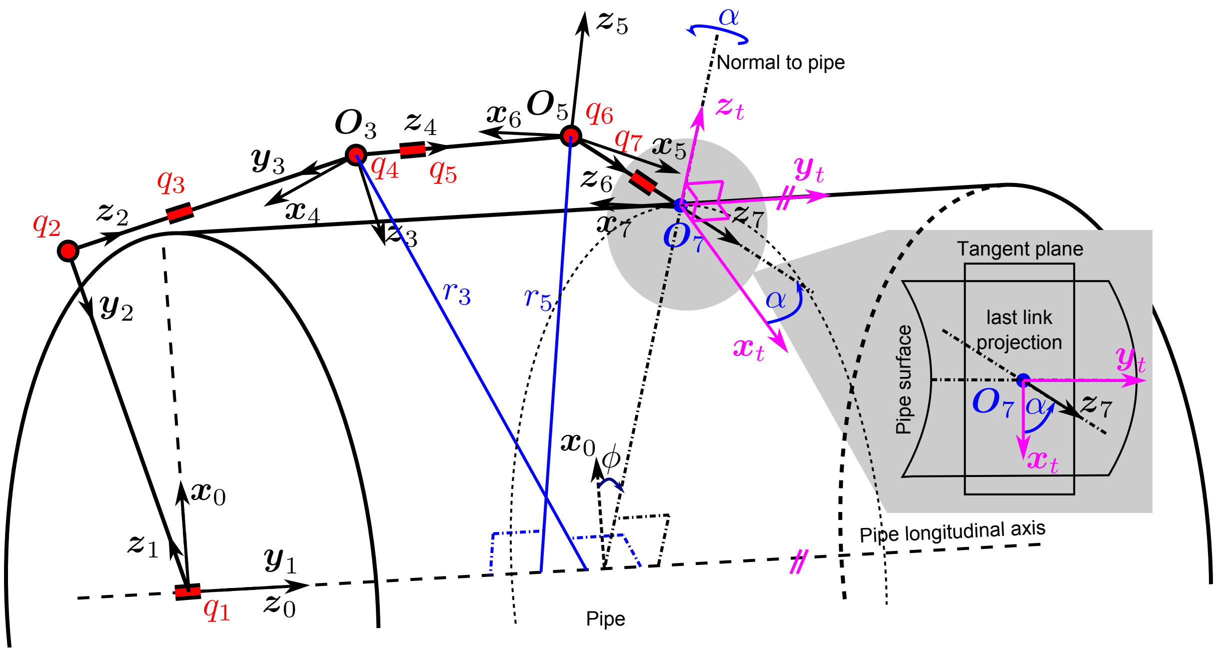

Figure 4: New generalized coordinates meaning

New generalized coordinates:

$$\boldsymbol{X} = \begin{bmatrix} x & y & z & r_3 & r_5 & \alpha \end{bmatrix}^T, \quad \boldsymbol{O}_7^0 = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{13}$$

Direct geometric model for new defined generalized coordinates can be computed as follows:

♦ Algorithm 2 (DGM for ROBIN robot - new generalized coordinates)

Algorithm input:

Joint coordinates:

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T, \quad q_7 = 0$$

Algorithm outputs:

Generalized coordinates:

$$\boldsymbol{X} = \begin{bmatrix} x & y & z & r_3 & r_5 & \alpha \end{bmatrix}^T$$

Algorithm parameters:

Length of the links:

$$\boldsymbol{\xi} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T$$

Layout of the algorithm (without implementation details):

1. Position $\boldsymbol{O}_7^0$ of the end-effector - the same as in (3):

$$
\begin{aligned}
&\boldsymbol{O}_7^0 = \begin{bmatrix} x & y & z \end{bmatrix}^T \\
&\quad x =((((C_1C_2C_3 - S_1S_3)C_4 + C_1S_2S_4)C_5 + (-C_1C_2S_3 - S_1C_3)S_5)S_6 - (-(C_1C_2C_3 - S_1S_3)S_4 + \\
&\qquad + C_1S_2C_4)C_6)L_4 + ((C_1C_2C_3 - S_1S_3)S_4 - C_1S_2C_4)L_3 - C_1S_2L_2 + S_1L_1 \\
&\quad y =((((S_1C_2C_3 + C_1S_3)C_4 + S_1S_2S_4)C_5 + (-S_1C_2S_3 + C_1C_3)S_5)S_6 - (-(S_1C_2C_3 + C_1S_3)S_4 + \\
&\qquad + S_1S_2C_4)C_6)L_4 + ((S_1C_2C_3 + C_1S_3)S_4 - S_1S_2C_4)L_3 - S_1S_2L_2 - C_1L_1 \\
&\quad z =(((S_2C_3C_4 - C_2S_4)C_5 - S_2S_3S_5)S_6 - (-S_2C_3S_4 - C_2C_4)C_6)L_4 + \\
&\qquad + (S_2C_3S_4 + C_2C_4)L_3 + C_2L_2)
\end{aligned}
\tag{14}
$$

2. 1st elbow distance r_3 (dependent only on q_2):

$$
r_3 = \sqrt{\left(\boldsymbol{O}_3^0[1:2]\right)^2 \boldsymbol{O}_3^0[1:2]} = \sqrt{L_1^2 + L_2^2 - L_2^2C_2^2} \tag{15}
$$

3. 2nd elbow distance r_5:

$$
\begin{aligned}
r_5 = \big(&-2L_3S_4C_2C_3S_2L_2 - 2L_3^2S_4C_2C_3S_2C_4 + L_3^2 + L_2^2 + L_1^2 - L_2^2C_2^2 - L_3^2C_3^2 - \\
&-L_3^2C_2^2C_3^2C_4^2 - 2L_3C_4L_2C_2^2 + L_3^2C_4^2C_3^2 - L_3^2C_4^2C_2^2 + L_3^2C_2^2C_3^2 + 2L_3C_4L_2 - 2L_3S_4S_3L_1\big)^{\frac{1}{2}}
\end{aligned}
\tag{16}
$$

4. The last robot link orientation α:

Tangent plane to pipe surface at point $\boldsymbol{O}_7^0$ is given by the axis $\boldsymbol{x}_t$, $\boldsymbol{y}_t$ of the CS F_t, see Figure 4, where rotation matrix $\boldsymbol{R}_t^0$ is:

$$
\begin{aligned}
&\boldsymbol{R}_t^0 = \begin{bmatrix} \boldsymbol{x}_t^0 & \boldsymbol{y}_t^0 & \boldsymbol{z}_t^0 \end{bmatrix}, \quad \text{kde:} \quad \boldsymbol{x}_t^0 = \begin{bmatrix} -S_\phi \\ C_\phi \\ 0 \end{bmatrix}, \quad \boldsymbol{z}_t^0 = \begin{bmatrix} C_\phi \\ S_\phi \\ 0 \end{bmatrix}, \quad \boldsymbol{y}_t^0 = \boldsymbol{z}_t^0 \times \boldsymbol{x}_t^0 \\
&\quad \phi = \operatorname{atan2}(\boldsymbol{O}_7^0[2], \boldsymbol{O}_7^0[1])
\end{aligned}
\tag{17}
$$

From known vector of $\boldsymbol{z}_7^0 = \boldsymbol{R}_7^0[1:3,3]$, see (3),

$$
\begin{aligned}
\boldsymbol{z}_7^0 &= \begin{bmatrix} \boldsymbol{z}_7^0[1] & \boldsymbol{z}_7^0[2] & \boldsymbol{z}_7^0[3] \end{bmatrix}^T \\
\boldsymbol{z}_7^0[1] &=(((C_5C_3C_4 - S_3S_5)C_2 + C_5S_2S_4)S_6 - C_6(S_2C_4 - C_3C_2S_4))C_1 - \\
&\quad - ((S_5C_3 + C_5C_4S_3)S_6 + C_6S_4S_3)S_1 \\
\boldsymbol{z}_7^0[2] &=(((C_5C_3C_4 - S_3S_5)C_2 + C_5S_2S_4)S_6 - C_6(S_2C_4 - C_3C_2S_4))S_1 + \\
&\quad + C_1((S_5C_3 + C_5C_4S_3)S_6 + C_6S_4S_3) \\
\boldsymbol{z}_7^0[2] &=((C_5C_3C_4 - S_3S_5)S_6 + C_6C_3S_4)S_2 + C_2(C_6C_4 - S_4C_5S_6))
\end{aligned}
\tag{18}
$$

the orientation of the last link (projection to plane $\boldsymbol{x}_t$, $\boldsymbol{y}_t$) is given as:

$$
\begin{aligned}
\boldsymbol{z}_7^t &= \left(\boldsymbol{R}_t^0\right)^T \boldsymbol{z}_7^0 \quad \text{(with respect to CS } F_t\text{)} \\
\alpha &= \operatorname{atan2}\left(\boldsymbol{z}_7^t[2], \boldsymbol{z}_7^t[1]\right)
\end{aligned}
\tag{19}
$$

♦

Unfortunately **Inverse gemetric model** for new defined generalized coordinates brings fundamental difficulties and the problem cannot be decomposed into two independent sub-problems as in the Algorithm 1 (*translation part* including q_2, q_3, q_4 computation, see points 3., 4., 5. and *rotation part* including q_5, q_6, q_7 computation, see point 6.).

For this reason the algorithm for IGM was based on standard numerical computation [3, 11] resulting in iterative algorithm for finding the DGM inverse problem. The algorithm can be summarized as follows:

♦ **Algorithm 3 (IGM for ROBIN robot - new generalized coordinates)**

Algorithm input:

Generalized coordinates:

$$\boldsymbol{X} = \begin{bmatrix} x & y & z & r_3 & r_5 & \alpha \end{bmatrix}^T$$

Robot initial state (joints position):

$$\boldsymbol{Q}^{init} = \begin{bmatrix} q_1^{init} & q_2^{init} & q_3^{init} & q_4^{init} & q_5^{init} & q_6^{init} \end{bmatrix}^T$$

Algorithm outputs:

Joint coordinates:

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T, \quad q_7 = 0$$

Algorithm parameters:

Length of the links:

$$\boldsymbol{\xi} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T$$

Layout of the algorithm (without implementation details):

1. Direct (analytical) computation of q_2 joint coordinate:

$$\begin{aligned} C_2 &= \begin{cases} +\sqrt{\frac{-r_3^2+L_1^2+L_2^2}{L_2}} & \text{for} \quad \cos(q_2^{init}) \geq 0 \\ -\sqrt{\frac{-r_3^2+L_1^2+L_2^2}{L_2}} & \text{for} \quad \cos(q_2^{init}) < 0 \end{cases} \\ S_2 &= \begin{cases} +\sqrt{1-C_2^2} & \text{for} \quad \sin(q_2^{init}) \geq 0 \\ -\sqrt{1-C_2^2} & \text{for} \quad \sin(q_2^{init}) < 0 \end{cases} \end{aligned} \tag{20}$$

2. Numeric computation of remaining joint coordinates (the inverse of following nonlinear vector equation - modified DGM without q_2 - r_3):

$$\boldsymbol{X}_{12356} = \mathbf{F}(\boldsymbol{Q}_{13456}) \in \mathbb{R}^5 \tag{21}$$

$$\text{gradient:} \quad \boldsymbol{G}\left(\boldsymbol{Q}_{13456}\right) = \frac{\partial \mathbf{F}(\boldsymbol{Q}_{13456})}{\partial \boldsymbol{Q}_{13456}} \in \mathbb{R}^{5,5} \tag{22}$$

$$\boldsymbol{Q}_{13456} = \begin{bmatrix} q_1 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T, \quad \boldsymbol{X}_{12356} = \begin{bmatrix} x & y & z & r_5 & \alpha \end{bmatrix}^T$$

Partial step of the iterative algorithm (Newton-Rhapson algorithm) is defined as $(k = 0, 1, \dots)$:

$$
\begin{aligned}
\text{Initial step:}\quad & \boldsymbol{Q}_{13456}^{(0)} = \boldsymbol{Q}^{init}[1,3:6] \\
\text{Iterate:}\quad & \boldsymbol{Q}_{13456}^{(k+1)} = \boldsymbol{Q}_{13456}^{(k)} + \Delta\boldsymbol{Q}_{13456}^{(k)} \\
& \Delta\boldsymbol{Q}_{13456}^{(k)} = k \cdot \boldsymbol{G}^{-1}\left(\boldsymbol{Q}_{13456}^{(k)}\right) \cdot \left(\boldsymbol{X}_{12356} - \mathbf{F}\left(\boldsymbol{Q}_{13456}^{(k)}\right)\right) \\
\text{Stopping cond.:}\quad & e < e_{max} \\
& e = \left(\boldsymbol{X}_{12356} - \mathbf{F}\left(\boldsymbol{Q}_{13456}^{(k)}\right)\right)^T \cdot \left(\boldsymbol{X}_{12356} - \mathbf{F}\left(\boldsymbol{Q}_{13456}^{(k)}\right)\right)
\end{aligned}
\tag{23}
$$

Note:

- The initial robot state $\boldsymbol{Q}^{init}$ can be computed from actual measurement of the actuator position after turning on of the robot or after switching mode of operation (e.g. standard and advanced IGM, see Algorithm 1, 3).
- A scalar constant k represents gain of the algorithm (line search gain), e_{max} represents max. algorithm error (stop condition).
- The solution of the IGM is given naturally from initial robot state (position) $\boldsymbol{Q}^{init}$.
- The gradient (jacobian matrix) $\boldsymbol{G}(\boldsymbol{Q}_{13456})$ can be derived analytically from $\mathbf{F}(\boldsymbol{Q}_{13456})$, but only one-step estimation is sufficient and much more robust than evaluation of analytic expression.

♦

4 Simulation results for advanced control algorithm

Robot virtual simulation model and advanced control algorithm, see Algorithm 3, was implemented in Matlab/Simulink/SimMechanics for the following kinematic parameters (links length):

$$\boldsymbol{\xi} = \begin{bmatrix} 0.4 & 0.3 & 0.3 & 0.3 \end{bmatrix}^T \ [m]$$

Constant position of the end-effector was chosen as:

$$\boldsymbol{X}[1:3] = \boldsymbol{O}_7^0 = \begin{bmatrix} 0.36 & 0 & 0.5612 \end{bmatrix}^T \ [m]$$

Initial robot state:

$$\boldsymbol{Q}^{init} = \begin{bmatrix} 0.6732 & -0.2162 & 0.4760 & 1.4627 & 2.8493 & 1.7390 & 1.9692 \end{bmatrix}^T \ [rad]$$

The following Figures show (for constant position of the end-effector):

- Figure 5: Three configurations of the 1st elbow distance
  (generalized coordinates $\boldsymbol{X}[4] = r_3$)
- Figure 6: Three configurations of the 2st elbow distance
  (generalized coordinates $\boldsymbol{X}[5] = r_5$)
- Figure 7: Three configurations of the last link orientation
  (generalized coordinates $\boldsymbol{X}[6] = \alpha$)

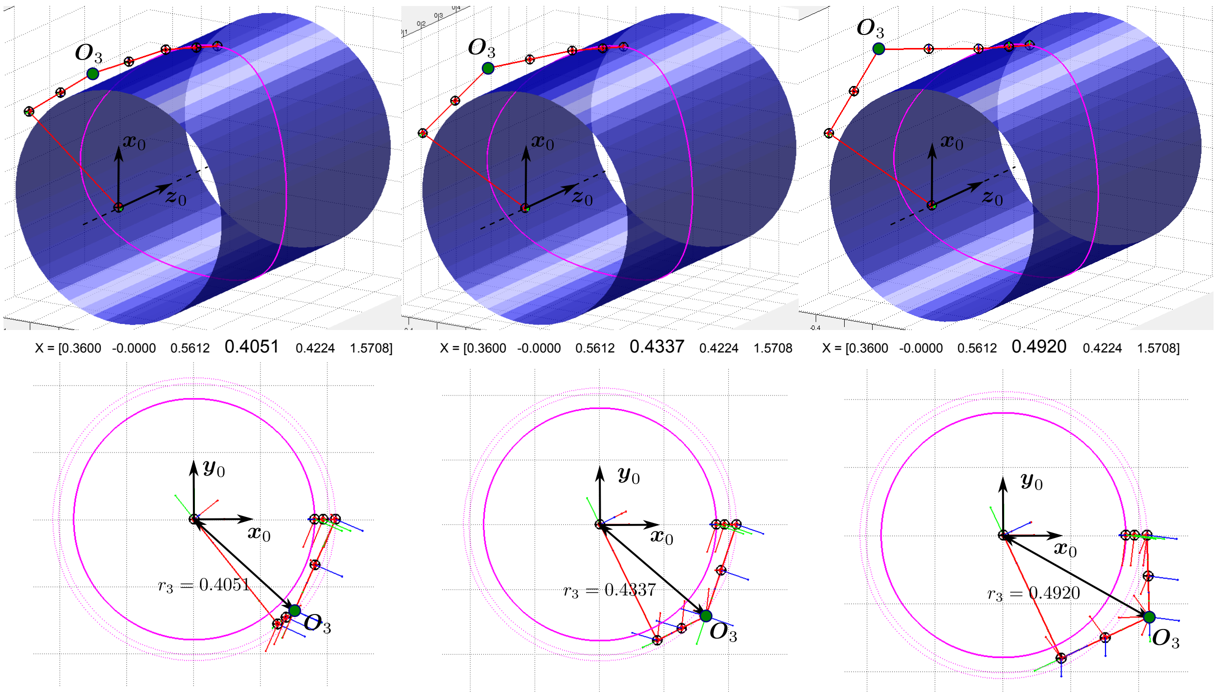

Figure 5: 1st elbow distance configuration: $r_3 = \{0.4051,\ 0.4337,\ 0.4920\}\ [m]$

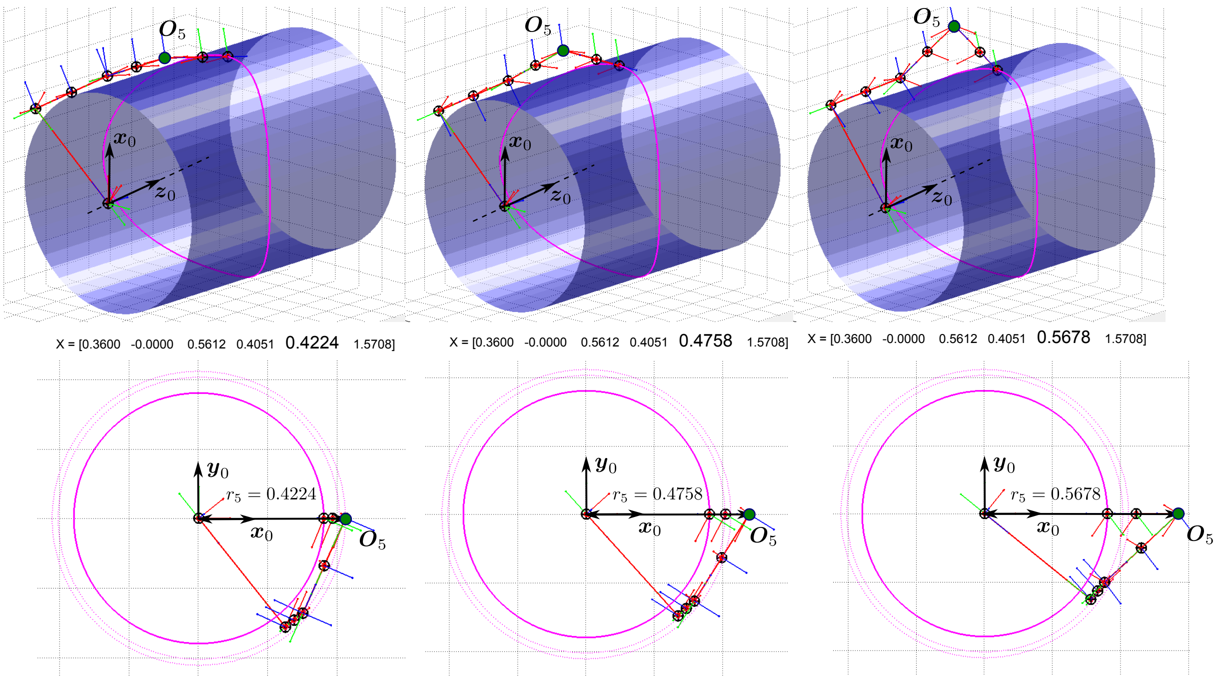

Figure 6: 2st elbow distance configuration: $r_5 = \{0.4224,\ 0.4758,\ 0.5678\}\ [m]$

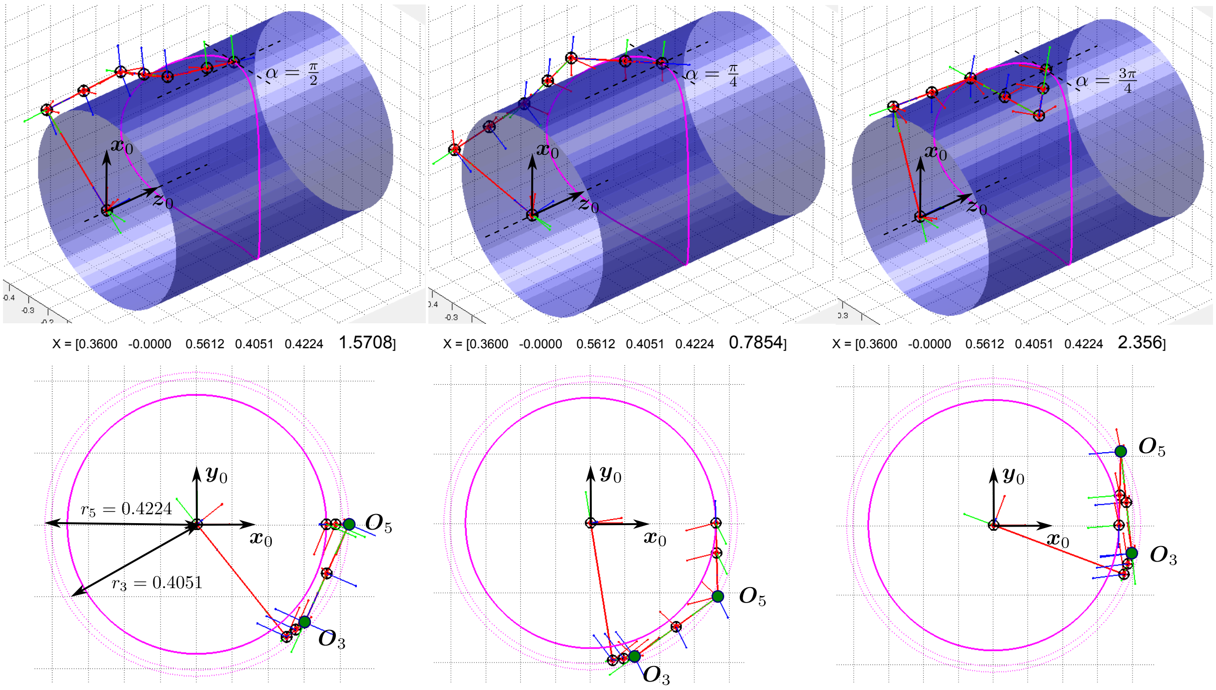

Figure 7: The last link orientation configuration: $\alpha = \{\frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}\}$ $[rad]$

5 conclusion

The paper was devoted to preliminary analysis of kinematic design and control algorithm of the first (conventional) part of the snake-like robot ROBIN. The advanced control algorithm was introduced for robot control in tasks concerning Non-destructive testing of pipe welds in order to control (minimize) the surrounding space occupied by the robot. The main advantage of the proposed algorithm is possibility to control the shape of robot body (especially distances of its elbows) while maintaining the position (translation) and orientation of the last link. The control algorithm for inverse kinematics computation was based on numeric iterative algorithm. The above examples show that on average 4 iterations per iterative step is sufficient for satisfactory accuracy (stopping condition $e_{max} = 1 \times 10^{-4}$).

Acknowledgement

This research was supported by project No. TF02000041 of the Technology Agency of the Czech Republic.

References

[1] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans. of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955. URL: http://ci.nii.ac.jp/naid/10008019314/en/.

[2] Martin Goubej, Martin Švejda, and Miloš Schlegel. *Úvod do mechatroniky, robotiky a systémů řízení pohybu (skripta)*. Zpadočeská univerzita v Plzni, 2012. URL: http://home.zcu.cz/~msvejda/URM/materialy/Uvod%20do%20mechatroniky.pdf.

[3] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing. Springer London, 2000. URL: http://books.google.fr/books?id=v9PLbcYd9aUC.

[4] M. Svejda and T. Cechura. Interpolation method for robot trajectory planning. In *Process Control (PC), 2015 20th International Conference on*, pages 406–411, June 2015. doi:10.1109/PC.2015.7169997.

[5] Martin Toupal. Řízení paralelního manipulátoru s nitinolovými (flexinolovými) aktuátory. Technical report, Západočeská univerzita v Plzni, Katedra kybernetiky, 2017.

[6] T. Čechura, A. Jáger, and M. Švejda. *Referenční příručka k robotu pro kontrolu svarů s omezenou přístupností*, 2016.

[7] Martin Švejda. Kinematika robotických architektur. Katedra Kybernetiky, ZČU v Plzni (online: http://home.zcu.cz/~msvejda/_publications/2011/rigo.pdf), 2011.

[8] Martin Švejda. Úvod do robotiky a mechatroniky. Přednášky k předmětu, 2012. URL: http://home.zcu.cz/~msvejda/URM/.

[9] Martin Švejda. New robotic architecture for ndt applications. In *World Congress IFAC*, volume 19, pages 11761–11766, 2014. doi:10.3182/20140824-6-ZA-1003.00989.

[10] Martin Švejda. Algoritmy plánování pohybu manipulátoru sáva. Technical report, Západočeská univerzita v Plzni, 2015. URL: http://home.zcu.cz/~msvejda/_publications/2015/3_AlgPlanovaniPohybu.pdf.

[11] Martin Švejda. *Optimalizace robotických architektur*. PhD thesis, Západočeská univerzita v Plzni, 2016. URL: http://home.zcu.cz/~msvejda/_publications/2016/4_SvejdaMartin_thesis_2016_06_14.pdf.