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

Project Number: TF02000041

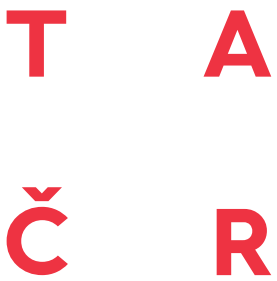
Virtual simulation models for ADRA-2I NDT robot (Czech pilot)

D5. (NTIS, ZSU) Virtual simulation model of robots - MIL

(Software)

Martin Švejda (ZCU)

20 August 2017



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

Abstract

The report deals with the virtual kinematic model of the ROBIN manipulator. The main work is devoted to the 6 DoF alternative part and its concatenation with 7 DoF conventional part of the robot. The virtual simulation model of the overall 13 DoF robot was introduced. The new algorithm for the optimal coordinated JOG of the end-effector regarding obstacles avoidance was analysed, implemented and tested. The main advantage stems from the possibility to control the robot end-effector motion in so-called developed view which corresponds to the standard pipeline system and appropriate welds documentation in power plants.

Contents

1 Conventional part of ROBIN robot

The 7 DoF conventional part of the ROBIN robot (1 DoF circumferential travel and 6 DoF serial manipulator with perpendicular axes of rotation), see Fig. 1, was studied in [10].

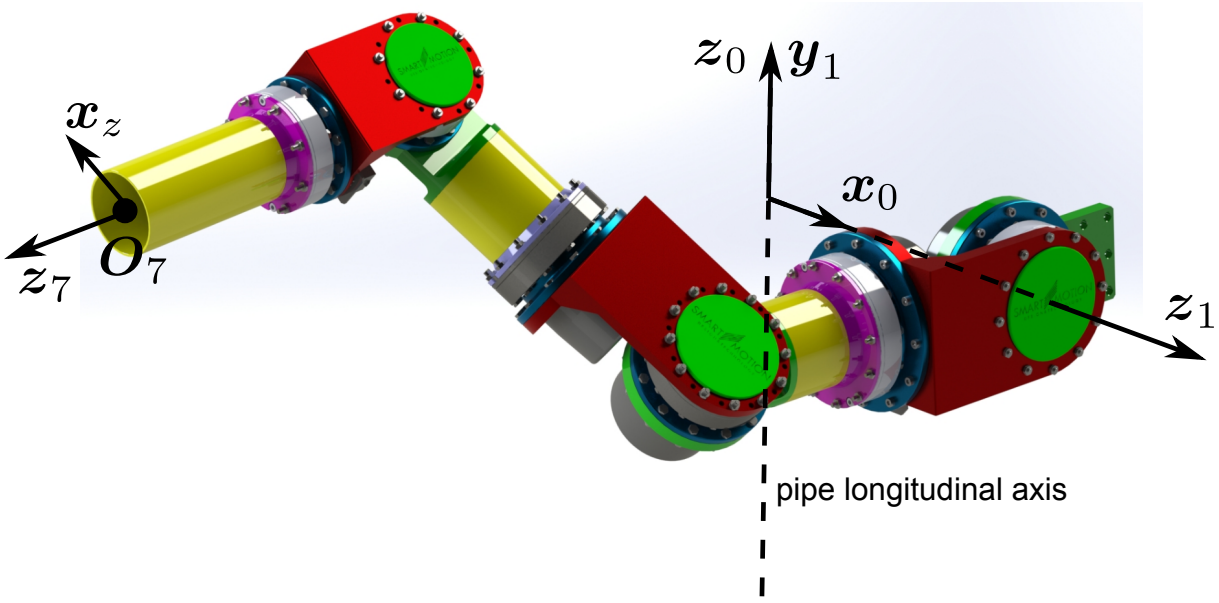


Figure 1: Conventional part of the ROBIN robot

The following problems were solved:

- Virtual simulation model in Matlab/Simulink/SimMechanics.
- Forward and inverse kinematic model - the full end-effector position and orientation control parametrized by the position of the circumferential travel (redundant joint coordinate)
- Alternative approach of the kinematic control where only the position $\boldsymbol{O}_7$ and orientation α of the last manipulator link projected to pipe surface were controlled. The remaining 2 DoF (the last joint q_7 was neglected) were used for control of the height of the extremal points $\boldsymbol{O}_3$, $\boldsymbol{O}_5$ above the pipe surface, see Fig. 2.

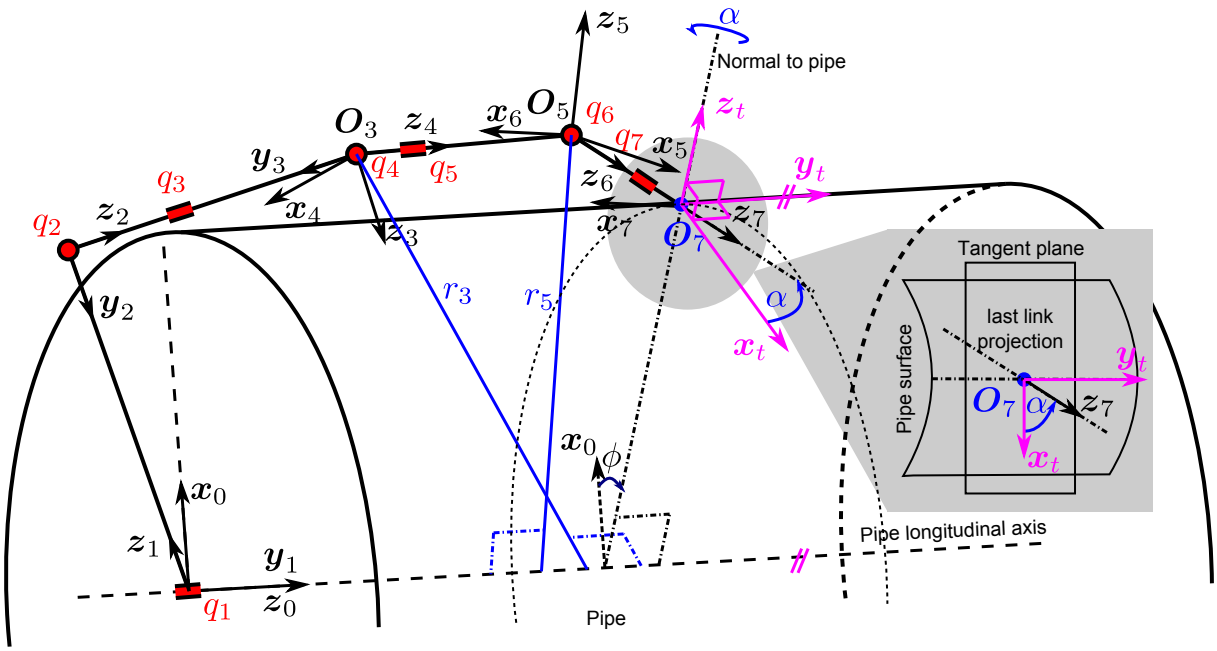
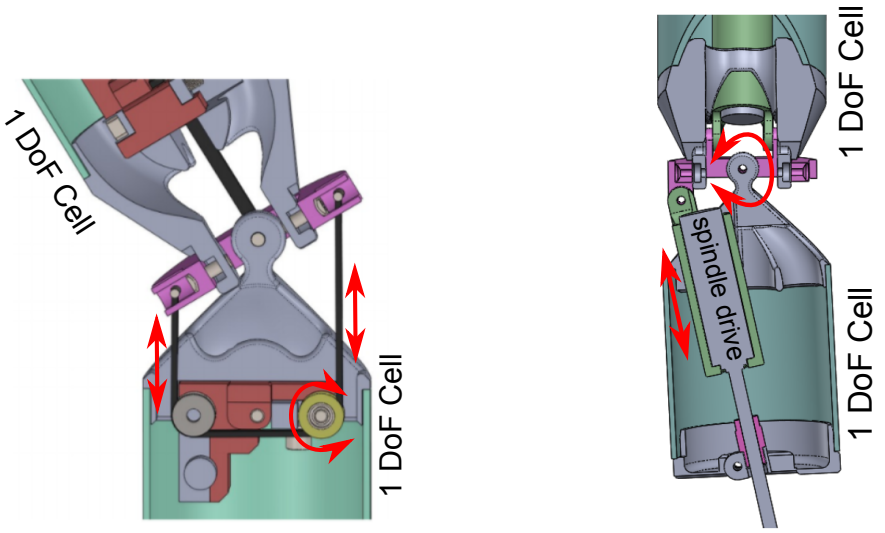


Figure 2: The generalized coordinates of the conventional part of the ROBIN robot

2 Alternative part of ROBIN robot

The alternative part of the ROBIN robot have been studied intensively in the previous works which were dedicated to the Smart Memory Alloy (SMA) systems for robot actuators [11, 4, 7] and the possibility to use micro-hydraulic systems for robot actuators [3]. Because of some difficulties which did not make possible to realize the proposed concepts of robot actuators the new and final actuators system have been introduced. The new idea is based on the simple 1 DoF actuator cell which is driven either by the toothed belt with micro DC motor or by the micro spindle drive with the DC motor, for details of the development, see [1]. The concept of two versions of actuator cells is shown in Fig. 3.



(a) Toothed belt driven actuator cell

(b) Spindle drive driven actuator cell

Figure 3: Two versions of the proposed actuator cells

2.1 Virtual simulation model

The virtual simulation model of the robot was designed in the Matlab/Simulink/SimMechanics and was based on the prerequisites (functions in Matlab and functional blocks in SimMechanics) which are a part of the library robotLib, see [9]. The manipulator consist of 3 2 DoF cells (where one 2 DoF cells is given by the interconnection of two simple 1 DoF cells with perpendicular axis of rotations - one rotation of each side). Therefore the 6 DoF serial kinematic chain is assembled with six mutually intersecting axes of rotation. The kinematics of the manipulator is given by the Denavit-Hartenberg (D-H) convention [2], the fixed coordinate systems (CSs) of the manipulator links are depicted in Fig. 4. D-H parameters are summarized in Table 1.

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

Table 1: D-H parameters of the manipulator

Joints coordinates are defined as:

$$\boldsymbol{Q} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 \end{bmatrix}^T \tag{1}$$

Generalized coordinates (end-effector position) are defined as (translation vector and rotation matrix):

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{O}_6^b & \boldsymbol{R}_6^b \end{bmatrix}$$

Kinematic parameters are defined as:

$$\boldsymbol{\xi} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \end{bmatrix}^T \quad \text{(Manipulator links lengths)} \tag{2}$$

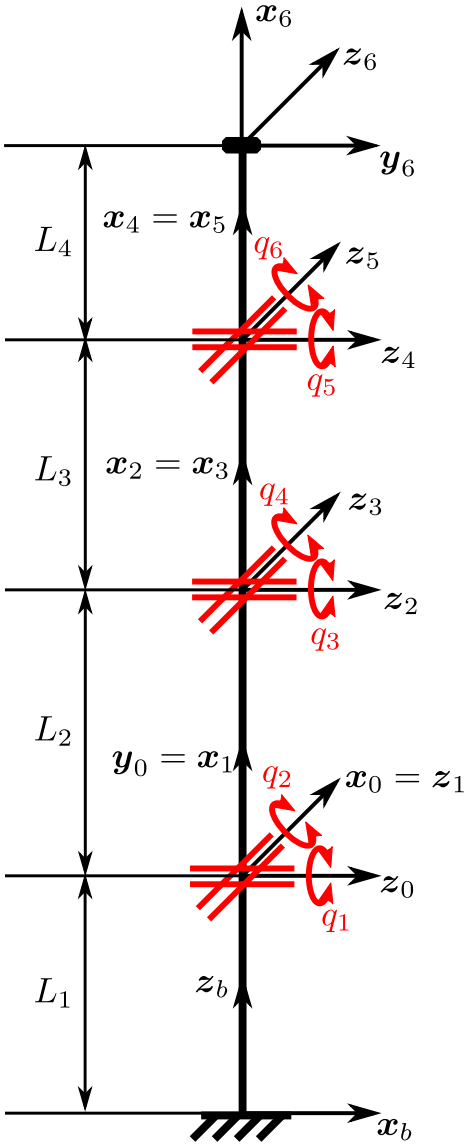


Figure 4: Coordinate systems according to D-H convention

$$\boldsymbol{T}_0^b = \left[\begin{array}{c:c} \boldsymbol{R}_0^b & \boldsymbol{O}_0^b \\ \hdashline 0 \quad 0 \quad 0 & 1 \end{array}\right] \quad \text{(Base compensation: translation vector and rotation matrix)} \quad (3)$$

Forward kinematic model

According to standard representation of CSs transformation the forward kinematic model of the manipulator is given by the multiplication of the homogeneous matrices as follows[1]:

[1]The notation of the point/vector/matrix $\boldsymbol{x}_i^j$ represents the point/vector/matrix $\boldsymbol{x}_i$ with respect to Coordinate System (CS) F_j. The notation $\boldsymbol{O}[a:b, c:d]$ represents selection of the a-th to b-th row and the c-th to d-th column of the matrix/vector $\boldsymbol{O}$.

$$\boldsymbol{T}_6^b = \boldsymbol{T}_b^0 \cdot \prod_{i=1}^{6} \boldsymbol{T}_i^{i-1}(d_i, \theta_i, a_i, \alpha_i), \quad \boldsymbol{T}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{O}_6^b = \boldsymbol{T}_6^b[1:3,4], \quad \boldsymbol{R}_6^b = \boldsymbol{T}_6^b[1:3,1:3] \tag{4}$$

Forward instantaneous kinematics for velocities and accelerations is given as:

$$\begin{bmatrix} \dot{\boldsymbol{O}}_6^b \\ \boldsymbol{\omega}_6^b \end{bmatrix} = \boldsymbol{J}_6^b(\boldsymbol{Q}) \cdot \dot{\boldsymbol{Q}}, \quad \begin{bmatrix} \ddot{\boldsymbol{O}}_6^b \\ \dot{\boldsymbol{\omega}}_6^b \end{bmatrix} = \dot{\boldsymbol{J}}_6^b(\boldsymbol{Q}, \boldsymbol{Q}) \cdot \dot{\boldsymbol{Q}} + \boldsymbol{J}_6^b(\boldsymbol{Q}) \cdot \ddot{\boldsymbol{Q}} \tag{5}$$

where $\boldsymbol{J}_6^b(\boldsymbol{Q})$ resp. $\dot{\boldsymbol{J}}_6^b(\dot{\boldsymbol{Q}}, \boldsymbol{Q})$ is kinematic jacobian resp. its time derivative.

Inverse kinematic model

Generally the inverse kinematic problem is much more difficult for serial manipulators and it can be shown that there are only some specific robot architectures where inverse kinematics can be solved analytically. In such case the methods for inverse kinematic computations often use e.g. kinematic decomposition which leads to a possibility to split the problem to partial sub-problems and to solve these problems separately. The well-know robot architectures (e.g. the most industrial manipulators) which fulfil the requirements on kinematic decomposition are the robots which contains spherical wrist (the rotation axis of the last three joints are perpendicular to each other).

Unfortunately, the proposed kinematic architecture of the manipulator can not be simply decomposed. Therefore the numeric algorithm for inverse kinematic is convenient to use. Moreover, in the case of kinematic redundancy some additional optimization may be included in the numeric algorithm. The kinematic redundancy becomes advantageous for our concept of NDT robot because some generalized coordinates need not necessarily be controlled. Typically only directions of the $\boldsymbol{x}_6$ axis is supposed to be controlled independently of its axial rotation, see Fig. 5. The new generalized coordinates (position) is given as:

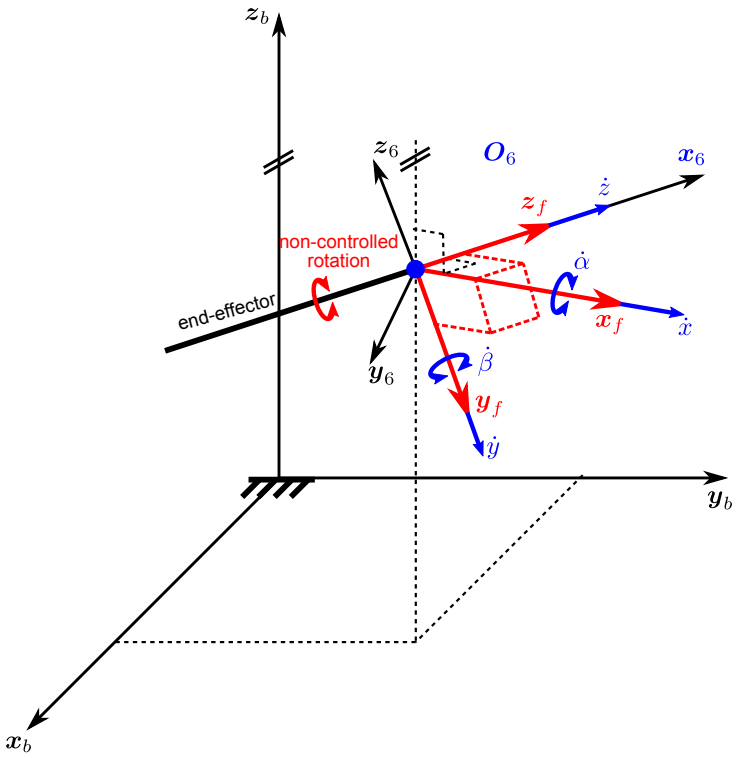


Figure 5: New generalized coordinates definition

$$\bar{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{O}_6^b \\ \boldsymbol{x}_6^b \end{bmatrix} \tag{6}$$

There are only 5 DoF of the end-effector ($\|\boldsymbol{x}_6^b\| = 1$).

In order to intuitive robot navigation by the operator the JOG function w.r.t. actual position of CS F_f (moving with actual position of the and-effector), see Fig. 5, the demanded end-effector velocities have to be recomputed as follows:

$$\begin{bmatrix} \dot{\boldsymbol{O}}_f^f \\ \boldsymbol{\omega}_f^f \end{bmatrix} = \begin{bmatrix} (\boldsymbol{R}_f^b)^T & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{R}_f^b)^T \end{bmatrix} \cdot \begin{bmatrix} \dot{\boldsymbol{O}}_f^b \\ \boldsymbol{\omega}_f^b \end{bmatrix} \tag{7}$$

and therefore $\boldsymbol{r}_{6,f}^6 = \boldsymbol{O}_f^6 = \boldsymbol{0}$ it holds: $\dot{\boldsymbol{O}}_f^b = \dot{\boldsymbol{O}}_e^b$ (and $\boldsymbol{\omega}_f^b = \boldsymbol{\omega}_e^b$):

$$\begin{bmatrix} \dot{\boldsymbol{O}}_f^f \\ \boldsymbol{\omega}_f^f \end{bmatrix} = \begin{bmatrix} (\boldsymbol{R}_f^b)^T & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{R}_f^b)^T \end{bmatrix} \cdot \begin{bmatrix} \dot{\boldsymbol{O}}_e^b \\ \boldsymbol{\omega}_e^b \end{bmatrix} = \underbrace{\begin{bmatrix} (\boldsymbol{R}_f^b)^T & \boldsymbol{0} \\ \boldsymbol{0} & (\boldsymbol{R}_f^b)^T \end{bmatrix} \cdot \boldsymbol{J}(\boldsymbol{Q})}_{\bar{\boldsymbol{J}}(\boldsymbol{Q})} \cdot \dot{\boldsymbol{Q}} \tag{8}$$

And rotation matrix $\boldsymbol{R}_f^6$ depends on actual manipulator position:

$$\boldsymbol{R}_f^b = \begin{bmatrix} \boldsymbol{x}_f^b & \boldsymbol{y}_f^b & \boldsymbol{z}_f^b \end{bmatrix}, \quad \boldsymbol{x}_f^b = \boldsymbol{R}_6^b(:,1) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \; \boldsymbol{y}_f^b = \boldsymbol{R}_6^b(:,1) \times \boldsymbol{x}_f^b, \; \boldsymbol{z}_f^b = \boldsymbol{x}_f^b \times \boldsymbol{y}_f^b \tag{9}$$

where $\boldsymbol{R}_6^b$ is rotation matrix from forward kinematics (4) and $\bar{\boldsymbol{J}}\boldsymbol{Q}$ is new kinematic jacobian.

The new forward instantaneous kinematics for redundant manipulator is given as, see Fig. 5:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \bar{\boldsymbol{J}}(\boldsymbol{Q})[1:5,:] \cdot \dot{\boldsymbol{Q}} \tag{10}$$

where $\dot{x}$, $\dot{y}$, $\dot{z}$ are translation velocities and $\dot{\alpha}$, $\dot{\beta}$ are rotation velocities w.r.t. actual CS F_f. The rotation along the $\boldsymbol{z}_f$ axis is non-controlled.

The well-known algorithm for joint velocities computation is given, see [6, 5, 8]:

$$\dot{\boldsymbol{Q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{Q}) \cdot \dot{\bar{\boldsymbol{X}}} + \left(\boldsymbol{I} - \boldsymbol{J}^{\dagger}(\boldsymbol{Q}) \cdot \boldsymbol{J}(\boldsymbol{Q}) \right) \cdot \frac{\partial w(\boldsymbol{Q})}{\partial \boldsymbol{Q}} \tag{11}$$

where $\boldsymbol{J}^{\dagger}(\boldsymbol{Q}) = \bar{\boldsymbol{J}}^{T}(\boldsymbol{Q}) \cdot \left(\bar{\boldsymbol{J}}(\boldsymbol{Q}) \cdot \bar{\boldsymbol{J}}^{T}(\boldsymbol{Q}) \right)^{-1}$ is a Penrose inverse of the jacobian $\bar{\boldsymbol{J}}(\boldsymbol{Q})$.

The term $\frac{\partial w(\boldsymbol{Q})}{\partial \boldsymbol{Q}} = \dot{\boldsymbol{Q}}_0$ corresponds to the joint velocities which do not effect the end-effector velocity $\dot{\boldsymbol{X}}$ generated from computed joint velocities $\dot{\boldsymbol{Q}}$ (via the null-space of jacobian $\boldsymbol{J}$). Therefore the velocities $\dot{\boldsymbol{Q}}_0$ generate only the internal motion of the manipulator and because of these velocities match the gradient of the objective function $w(\boldsymbol{Q})$ the generated manipulator internal motion leads to the maximization of the objective function $w(\boldsymbol{Q})$. Note, that for $w(\boldsymbol{Q}) = 0$ the computed joint velocities $\dot{\boldsymbol{Q}}$ corresponding to the demanded end-effector velocity $\dot{\boldsymbol{X}}$ are minimized[2]. Therefore the optimal JOG is given as:

$$\dot{\boldsymbol{Q}} = \boldsymbol{J}^{\dagger}(\boldsymbol{Q}) \cdot \dot{\bar{\boldsymbol{X}}} \tag{12}$$

2.2 Control algorithm design

The control algorithm is based on the velocity dependencies (12) and it is realized in Matlab/SimMechanics. The demanded end-effector velocities $\dot{x}$, $\dot{y}$, $\dot{z}$, $\dot{\alpha}$, $\dot{\beta}$ are generated as keyboard inputs (via implemented user interface).

3 Overall robot

The main idea of the ADRA-2I project is to concatenate the first conventional part of ROBIN robot, see Chapter 1 with proposed alternative part, see Chapter 2. In that case we get the model of the model of 13 DoF serial manipulator. The following section describe the virtual model of the overall manipulator and the proposed control algorithm which is designed especially for robot motion control respecting standard methodology of pipe welds description.

3.1 Virtual simulation model

The kinematics of the manipulator is given by the D-H convention [2] and it corresponds with conventional and alternative parts concatenation, see Fig. 6. D-H parameters are summarized in Table 2.

[2]The algorithm (11) can be extended by the position error therm which ensures that demanded position of the end-effector is asymptotically reached. It is not necessary in the case under consideration where only manipulator JOG is used (position feedback is performed by the operator).

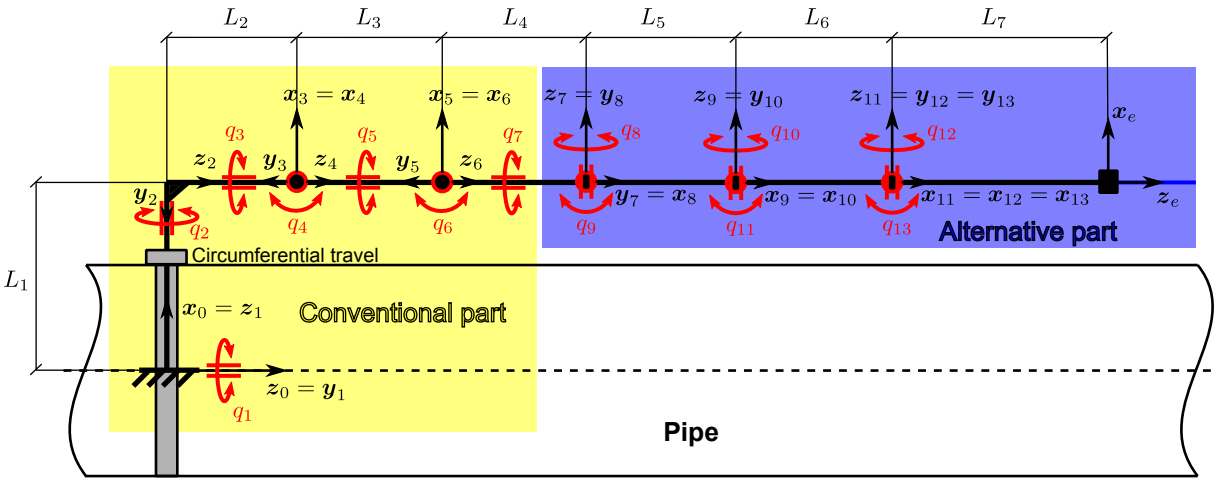


Figure 6: Coordinate systems of overall manipulator according to D-H convention

i	d_i	θ_i	a_i	α_i
1	0	θ_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	$\frac{\pi}{2}$
8	0	θ_8	0	$\frac{\pi}{2}$
9	0	θ_9	L_5	$-\frac{\pi}{2}$
10	0	θ_{10}	0	$\frac{\pi}{2}$
11	0	θ_{11}	L_6	$-\frac{\pi}{2}$
12	0	θ_{12}	0	$\frac{\pi}{2}$
13	0	θ_{13}	0	0

Table 2: D-H parameters of overall manipulator

Joints coordinates are defined as:

$$\boldsymbol{Q} = \begin{bmatrix} \theta_1 & \theta_2 & \dots & \theta_{12} & \theta_{13} \end{bmatrix}^T \tag{13}$$

Generalized coordinates (end-effector position) are defined as (translation vector and rotation matrix):

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{O}_e^0 & \boldsymbol{R}_e^0 \end{bmatrix}$$

Kinematic parameters are defined as:

$$\boldsymbol{\xi} = \begin{bmatrix} L_1 & L_2 & \dots & L_7 \end{bmatrix}^T \quad \text{(Manipulator links lengths)} \tag{14}$$

$$\boldsymbol{T}_e^{13} = \begin{bmatrix} 0 & 0 & 1 & L_7 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(End-effector compensation: translation vector and rotation matrix)} \tag{15}$$

Forward kinematic model

The standard forward kinematics model is implemented in the same way as above:

$$\boldsymbol{T}_e^0 = \prod_{i=1}^{13} \boldsymbol{T}_i^{i-1}(d_i, \theta_i, a_i, \alpha_i) \cdot \boldsymbol{T}_e^{13}, \quad \boldsymbol{T}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{O}_e^0 = \boldsymbol{T}_e^0[1:3, 4], \quad \boldsymbol{R}_e^0 = \boldsymbol{T}_e^0[1:3, 1:3] \tag{16}$$

And the standard forward instantaneous kinematics for velocities and accelerations is given as:

$$\begin{bmatrix} \dot{\boldsymbol{O}}_e^0 \\ \boldsymbol{\omega}_e^0 \end{bmatrix} = \boldsymbol{J}_e^0(\boldsymbol{Q}) \cdot \dot{\boldsymbol{Q}}, \quad \begin{bmatrix} \ddot{\boldsymbol{O}}_e^0 \\ \dot{\boldsymbol{\omega}}_e^0 \end{bmatrix} = \dot{\boldsymbol{J}}_e^0(\boldsymbol{Q}, \boldsymbol{Q}) \cdot \dot{\boldsymbol{Q}} + \boldsymbol{J}_e^0(\boldsymbol{Q}) \cdot \ddot{\boldsymbol{Q}} \tag{17}$$

where $\boldsymbol{J}_e^0(\boldsymbol{Q})$ resp. $\dot{\boldsymbol{J}}_e^0(\dot{\boldsymbol{Q}}, \boldsymbol{Q})$ is kinematic jacobian resp. its time derivative.

Standard description of the forward kinematics (16, 17) is not suitable for trajectory planning of NDT robot because of non-intuitive generalized coordinates representation (6 DoF movement in space). Instead of the standard representation the new generalized coordinates are introduced. The new generalized coordinates project the standard 6DoF coordinates on the pipe surface (analogously as cylindrical coordinates transformation). In this developed view only four generalized coordinates are defined and controlled because of the remaining DoF which are necessary for NDT probe positioning can be ensured by passive probe holder for pipe surface contact with an additional 1 rotation DoF for probe heading. The probe holder including its active heading are not further discussed. The proposed new four generalized coordinates

$$\bar{\boldsymbol{X}} = \begin{bmatrix} l & z & h & \alpha \end{bmatrix}^T \tag{18}$$

define only the position of the entire holder where l is the pipe perimeter, z is a long of pipe, h is a height above the pipe surface and α is the last manipulator link orientation projected on the pipe surface. Fig. 7 shows the developed view of pipe with two perpendicular tube inlets.

The transformation from standard generalized coordinates $\boldsymbol{X}$ (only translation $\boldsymbol{O}_e^0$ and the axis $\boldsymbol{z}_e^0$ - projection of the last link on the pipe surface $\Rightarrow$ orientation α) to the new generalized coordinates $\bar{\boldsymbol{X}}$ is given as:

$$\begin{bmatrix} \boldsymbol{O}_e^0 & \boldsymbol{R}_e^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{O}_e^0 & \boldsymbol{x}_e^0 & \boldsymbol{y}_e^0 & \boldsymbol{z}_e^0 \end{bmatrix} = \begin{bmatrix} O_x & x_x & y_x & z_x \\ O_y & x_y & y_y & z_y \\ O_z & x_z & y_z & z_z \end{bmatrix} \Rightarrow \text{Redefine: } \boldsymbol{X} = \begin{bmatrix} O_x \\ O_y \\ O_z \\ z_x \\ z_y \\ z_z \end{bmatrix}$$

$$\bar{\boldsymbol{X}} = \begin{bmatrix} l \\ z \\ h \\ \alpha \end{bmatrix} \tag{19}$$

$$\bar{\boldsymbol{X}} = \mathbf{F}(\boldsymbol{X}) = \begin{bmatrix} \arctan(O_y, O_x)\, R \\ O_z \\ -\frac{\sqrt{\left({O_y}^2 + {O_x}^2\right)^{-1}} R - 1}{\sqrt{\left({O_y}^2 + {O_x}^2\right)^{-1}}} \\ \arctan\left(z_z, -\frac{O_y\, z_x}{\sqrt{{O_y}^2 + {O_x}^2}} + \frac{O_x\, z_y}{\sqrt{{O_y}^2 + {O_x}^2}}\right) \end{bmatrix}$$

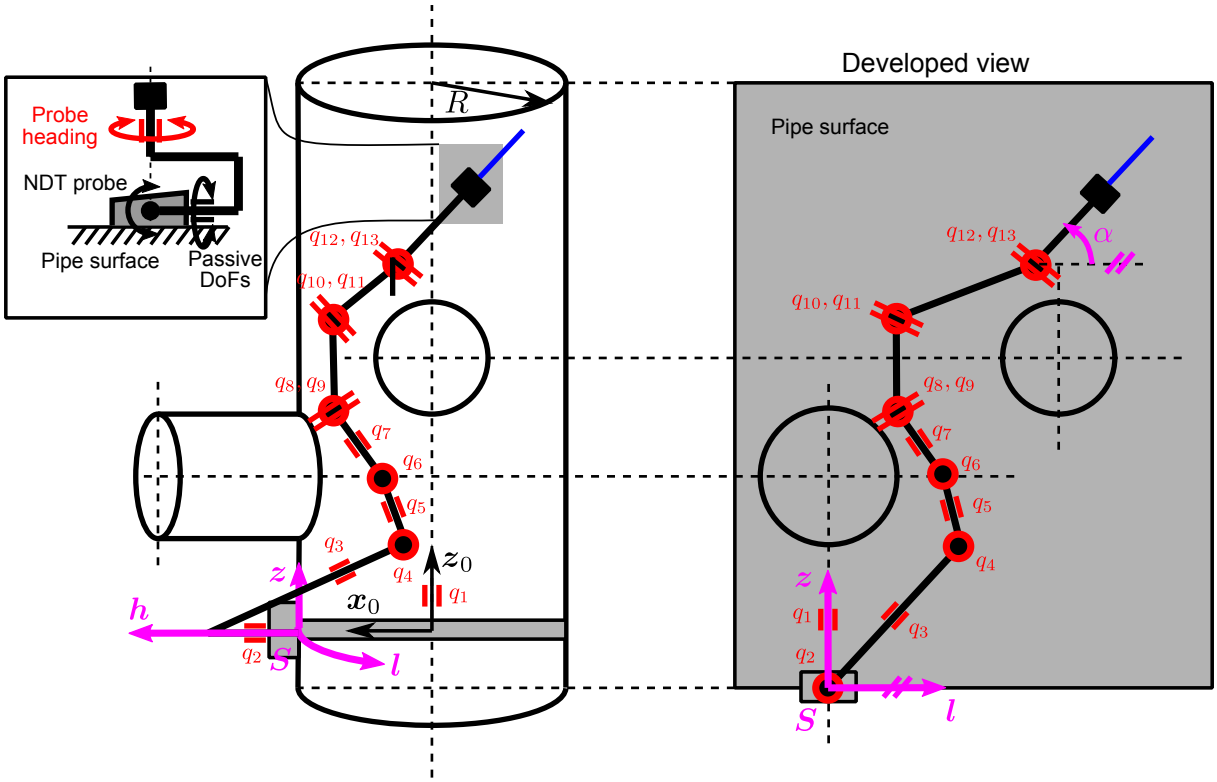


Figure 7: Developed view of the pipe surface

where R is the pipe radius.

The transformation between corresponding velocities of the standard and the new generalized coordinates can be expressed by the gradient $\frac{\partial \mathbf{F}(\boldsymbol{X})}{\partial \boldsymbol{X}}$:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{O}_x \\ \dot{O}_y \\ \dot{O}_z \\ \dot{z}_x \\ \dot{z}_y \\ \dot{z}_z \end{bmatrix}, \quad \bar{\boldsymbol{X}} \begin{bmatrix} \dot{l} \\ \dot{z} \\ \dot{h} \\ \dot{\alpha} \end{bmatrix} \tag{20}$$

$$\dot{\bar{\boldsymbol{X}}} = \frac{\partial \mathbf{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} \cdot \dot{\boldsymbol{X}}$$

$$\frac{\partial \mathbf{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{bmatrix} -\frac{r\,O_y}{{O_y}^2+{O_x}^2} & 0 & \frac{O_x}{\sqrt{({O_y}^2+{O_x}^2)^{-1}}({O_y}^2+{O_x}^2)} & -\frac{O_y\,(z_x\,O_x+z_y\,O_y)z_z}{\sqrt{{O_y}^2+{O_x}^2}({O_y}^2{z_x}^2-2\,O_y\,z_x\,O_x\,z_y+{O_x}^2{z_y}^2+{z_z}^2{O_y}^2+{z_z}^2{O_x}^2)} \\ \frac{r\,O_x}{{O_y}^2+{O_x}^2} & 0 & \frac{O_y}{\sqrt{({O_y}^2+{O_x}^2)^{-1}}({O_y}^2+{O_x}^2)} & \frac{O_x\,(z_x\,O_x+z_y\,O_y)z_z}{\sqrt{{O_y}^2+{O_x}^2}({O_y}^2{z_x}^2-2\,O_y\,z_x\,O_x\,z_y+{O_x}^2{z_y}^2+{z_z}^2{O_y}^2+{z_z}^2{O_x}^2)} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{O_y\sqrt{{O_y}^2+{O_x}^2}z_z}{{O_y}^2{z_x}^2-2\,O_y\,z_x\,O_x\,z_y+{O_x}^2{z_y}^2+{z_z}^2{O_y}^2+{z_z}^2{O_x}^2} \\ 0 & 0 & 0 & -\frac{O_x\sqrt{{O_y}^2+{O_x}^2}z_z}{{O_y}^2{z_x}^2-2\,O_y\,z_x\,O_x\,z_y+{O_x}^2{z_y}^2+{z_z}^2{O_y}^2+{z_z}^2{O_x}^2} \\ 0 & 0 & 0 & \frac{(-O_y\,z_x+O_x\,z_y)\sqrt{{O_y}^2+{O_x}^2}}{{O_y}^2{z_x}^2-2\,O_y\,z_x\,O_x\,z_y+{O_x}^2{z_y}^2+{z_z}^2{O_y}^2+{z_z}^2{O_x}^2} \end{bmatrix}^T$$

And the forward instantaneous kinematic leads to:

$$\dot{\bar{\boldsymbol{X}}} = \bar{\boldsymbol{J}}(\boldsymbol{Q}) \cdot \dot{\boldsymbol{Q}} \tag{21}$$

Where:

$$\bar{\boldsymbol{J}}(\boldsymbol{Q}) = \frac{\partial \mathbf{F}(\boldsymbol{X})}{\partial \boldsymbol{X}} \cdot \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{S}(\boldsymbol{z}_e^0) \end{bmatrix} \cdot \boldsymbol{J}_e^0(\boldsymbol{Q}), \quad \boldsymbol{S}(\boldsymbol{z}_e^0) = \begin{bmatrix} 0 & -z_z & z_y \\ z_z & 0 & -z_x \\ -z_y & z_x & 0 \end{bmatrix}$$

is the jacobian of the manipulator with respect to the new generalized coordinates $\bar{\boldsymbol{X}}$.

3.2 Control algorithm design

After redefinition of the forward kinematic problem (13 joint coordinates and only 4 generalized coordinates) it is clear that the proposed manipulator is redundant. Therefore the above mentioned numerical algorithm for inverse kinematic computation can be used. Because the NDT inspection is often supposed to be performed in very restricted area (hursh industrial environment) it is very convenient to use internal manipulator motion optimization which is based on suitable objective function $w(\boldsymbol{Q})$.

Despite of the common method for trajectory planning of hyper-redundant robots (in the sense of the tens of DoFs) where the robot body is enforced to follow the "robot head" movement the proposed idea of control algorithm is different because the exact head following is not necessary and number of the remaining redundant DoFs is limited (13 - 4 = 9 redundant DoFs). The objective function under consideration evaluates the distance among manipulator links and the obstacles and it is defined as follows, see Fig. 8:

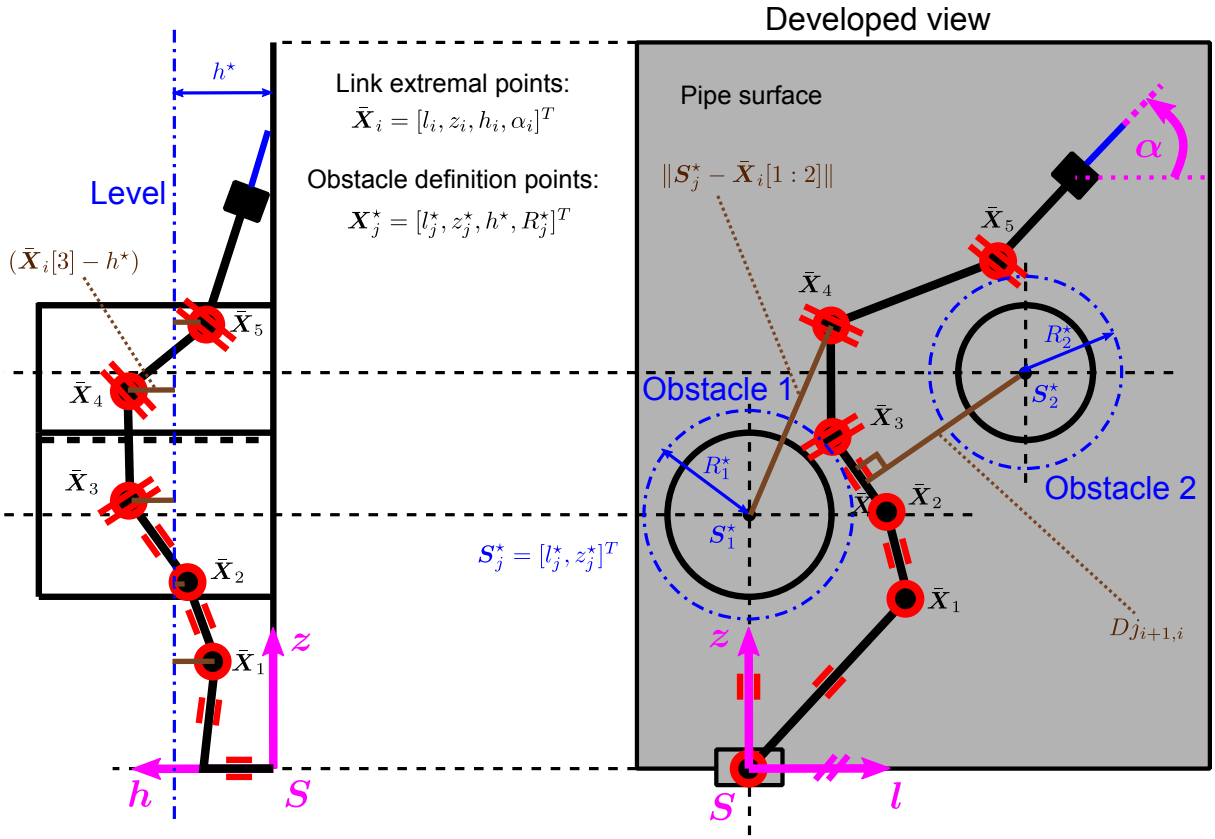


Figure 8: Objective function definition

The manipulator link extremal points are defined as follows:

$$\bar{\boldsymbol{X}}_i = \mathbf{F}(\boldsymbol{X}_k), \quad \boldsymbol{X}_k = \begin{bmatrix} \boldsymbol{O}_k^0 \\ \boldsymbol{z}_k^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_k^0[1:3,4] \\ \boldsymbol{T}_k^0[1:3,3] \end{bmatrix}, \quad i = \{1,2,3,4\},\ k = \{4,6,8,10,13\} \tag{22}$$

where $\boldsymbol{T}_k^0$ are homogeneous transformation matrices from the forward kinematics computation (16).

The objective function $w(\boldsymbol{Q})$ consists of the following contributions:

- Manipulator extremal points $\bar{\boldsymbol{X}}_i = [h_i]$ surface distance from $h^\star \Rightarrow w^2(\boldsymbol{Q})$

$$w^2(\boldsymbol{Q}) = -\sum_{i=1}^{5} \frac{1}{2} \cdot k_i^2 \cdot (\bar{\boldsymbol{X}}_i - h^\star)^2 \tag{23}$$

$$\frac{\partial w^2(\boldsymbol{Q})}{\partial \boldsymbol{Q}} = -\sum_{i=1}^{5} k_i^2 \cdot (\bar{\boldsymbol{X}}_i - h^\star) \cdot \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}} \tag{24}$$

- Manipulator extremal points $\bar{\boldsymbol{X}}_i = [l_i, z_i]^T$ distance from the center $\boldsymbol{S}_j^\star = [l_j^\star, z_j^\star]^T$ of the j-th circular obstacle with radius $R_j^\star \Rightarrow w_j^1(\boldsymbol{Q})$

$$w_j^1(\boldsymbol{Q}) = \sum_{i=1}^{5} \frac{1}{2} \cdot w_j^\star \cdot \hat{k}_i \cdot (\|\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i\|^2 - (R_j^\star)^2) \tag{25}$$

$$\hat{k}_i = \begin{cases} k_i^1 \;\; \text{for: } (\|\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i\|^2 - (R_j^\star)^2) < 0 \\ 0 \;\; \text{otherwise} \end{cases}$$

$$\frac{\partial w_j^2(\boldsymbol{Q})}{\partial \boldsymbol{Q}} = -\sum_{i=1}^{5} w_j^\star \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i)^T \cdot \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}} \tag{26}$$

- Manipulator link $[\bar{\boldsymbol{X}}_i = [l_{i+1}, z_{i+1}]^T, \bar{\boldsymbol{X}}_{i+1} = [l_i, z_i]^T]$ distance from the center $\boldsymbol{S}_j^\star = [l_j^\star, z_j^\star]^T$ of the j-th circular obstacle with radius $R_j^\star \Rightarrow w_j^3(\boldsymbol{Q})$

$$w_j^3(\boldsymbol{Q}) = \sum_{i=1}^{4} \frac{1}{2} \cdot w_j^\star \cdot \hat{k}_i \cdot (Dj_{i+1,i}^2(\boldsymbol{Q}) - (R_j^\star)^2) \tag{27}$$

$$\hat{k}_i = \begin{cases} k_i^1 \;\; \text{for: } (Dj_{i+1,i}^2(\boldsymbol{Q}) - (R_j^\star)^2) < 0 \;\wedge\; k_{i,i+1} \in \langle 0, 1 \rangle \\ 0 \;\; \text{otherwise} \end{cases}$$

$$\frac{\partial w_j^3(\boldsymbol{Q})}{\partial \boldsymbol{Q}} = \sum_{i=1}^{4} w_j^\star \cdot \hat{k}_i \cdot Dj_{i+1,i}(\boldsymbol{Q}) \cdot \frac{\partial Dj_{i+1,i}(\boldsymbol{Q})}{\partial \boldsymbol{Q}} \tag{28}$$

where

$$
\begin{aligned}
k_{i,i+1} &= \frac{(\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i)^T \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i)}{\|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|^2} \\
\bar{\boldsymbol{X}} &= \bar{\boldsymbol{X}}_i + k_{i,i+1} \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i) \\
Dj_{i+1,i}(\boldsymbol{Q}) &= \left[(\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}})^T \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}})\right]^{\frac{1}{2}} \\
\frac{\partial Dj_{i+1,i}(\boldsymbol{Q})}{\partial \boldsymbol{Q}} &= -\left[(\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}})^T \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}})\right]^{-\frac{1}{2}} \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}})^T \cdot \frac{\partial \bar{\boldsymbol{X}}}{\partial \boldsymbol{Q}} \\
\frac{\partial \bar{\boldsymbol{X}}}{\partial \boldsymbol{Q}} &= \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}} + \frac{\partial k_{i,i+1}}{\partial \boldsymbol{Q}} \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i) + k_{i,i+1} \cdot \left(\frac{\partial \bar{\boldsymbol{X}}_{i+1}}{\partial \boldsymbol{Q}} - \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}}\right) \\
\frac{\partial k_{i,i+1}}{\partial \boldsymbol{Q}} &= -2\|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|^{-1} \cdot \frac{\partial \|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|}{\partial \boldsymbol{Q}} \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i)^T \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i) - \\
&\quad - \|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|^{-2} \cdot \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}}^T \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i) + \\
&\quad + \|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|^{-2} \cdot (\boldsymbol{S}_j^\star - \bar{\boldsymbol{X}}_i)^T \cdot \left(\frac{\partial \bar{\boldsymbol{X}}_{i+1}}{\partial \boldsymbol{Q}} - \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}}\right) \\
\frac{\partial \|\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i\|}{\partial \boldsymbol{Q}} &= \left[(\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i)^T \cdot (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i)\right]^{-\frac{1}{2}} (\bar{\boldsymbol{X}}_{i+1} - \bar{\boldsymbol{X}}_i)^T \cdot \left(\frac{\partial \bar{\boldsymbol{X}}_{i+1}}{\partial \boldsymbol{Q}} - \frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}}\right)
\end{aligned}
$$

And it holds, see (21), for each extremal point:

$$
\frac{\partial \bar{\boldsymbol{X}}_i}{\partial \boldsymbol{Q}} = \frac{\partial \mathbf{F}(\boldsymbol{X}_k)}{\partial \boldsymbol{X}_k} \cdot \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & -\boldsymbol{S}(\boldsymbol{z}_k^0) \end{bmatrix} \cdot \boldsymbol{J}_k^0(\boldsymbol{Q}) \tag{29}
$$

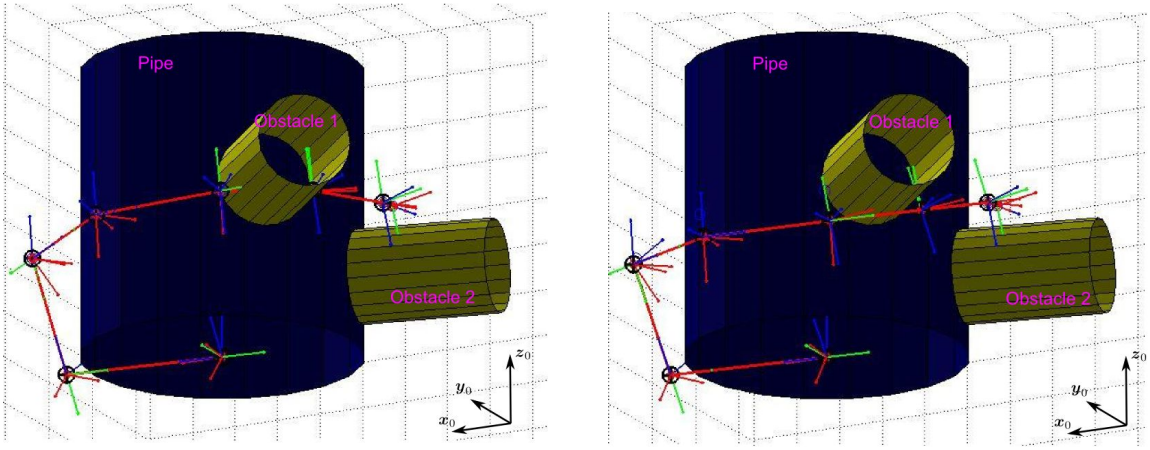
The constant k_i^1 resp. k_i^2 are the weights for the i-th extremal point distance and $[i, i+1]$-th manipulator link distance from obstacles resp. the i-th extremal point surface distance. The constants $w_j^\star$ are weights of the j-th obstacle.

The entire objective function result in:

$$
\begin{aligned}
w(\boldsymbol{Q}) &= \sum_j \left(w_j^1(\boldsymbol{Q}) + w_j^3(\boldsymbol{Q})\right) + w^2(\boldsymbol{Q}) \\
\frac{\partial w(\boldsymbol{Q})}{\partial \boldsymbol{Q}} &= \sum_j \left(\frac{\partial w_j^1(\boldsymbol{Q})}{\partial \boldsymbol{Q}} + \frac{\partial w_j^3(\boldsymbol{Q})}{\partial \boldsymbol{Q}}\right) + \frac{w^2(\boldsymbol{Q})}{\partial \boldsymbol{Q}}
\end{aligned} \tag{30}
$$

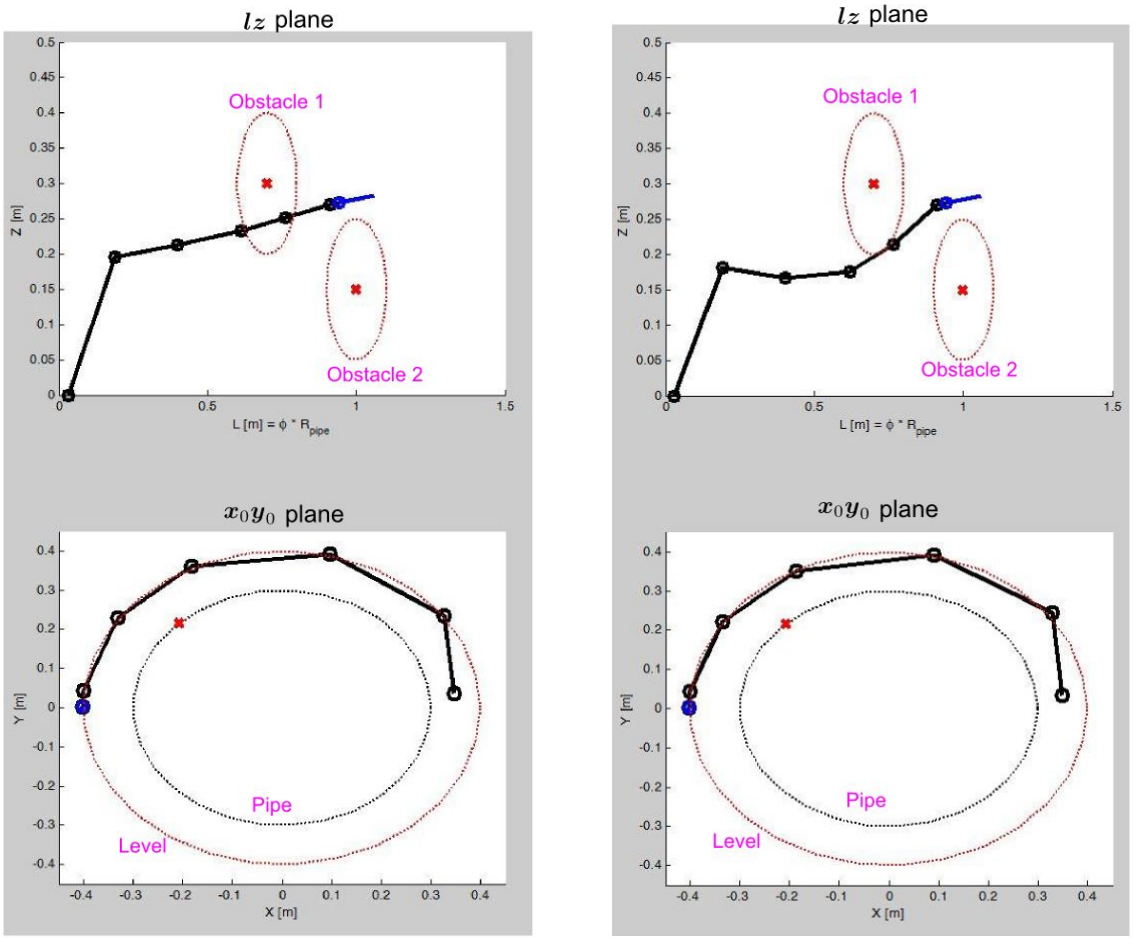
3.3 Simulation results

The inverse kinematic algorithm given by (11) with respect to the objective function (30) was demonstrated on the virtual simulation model of overall 13 DoF manipulator with a new generalized coordinates (19), see Chapter 3.1. The initial position of the manipulator (without optimization) is depicted in Fig. 9(a), 10(a) and the resulting position of the manipulator after its internal reconfiguration based on generated optimal joints velocities is depicted in Fig. 9(b), 10(b).



(a) Initial position of the manipulator (b) Resulting position of the manipulator

Figure 9: Overall redundant manipulator (13 joint coordinates / 4 generalized coordinates), initial position and resulting position after internal reconfiguration based on the proposed optimal algorithm, virtual simulation model in SimMechanics w.r.t. CS $\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{z}_0$



(a) Initial position of the manipulator (b) Resulting position of the manipulator

Figure 10: Overall redundant manipulator (13 joint coordinates / 4 generalized coordinates), initial position and resulting position after internal reconfiguration based on the proposed optimal algorithm, virtual simulation w.r.t. CS $\boldsymbol{l}, \boldsymbol{z}$ and CS $\boldsymbol{x}_0, \boldsymbol{y}_0$

The screen shots of the manipulator motion enforced by the JOG in the $\boldsymbol{l}$, $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ directions are depicted in Fig. 11

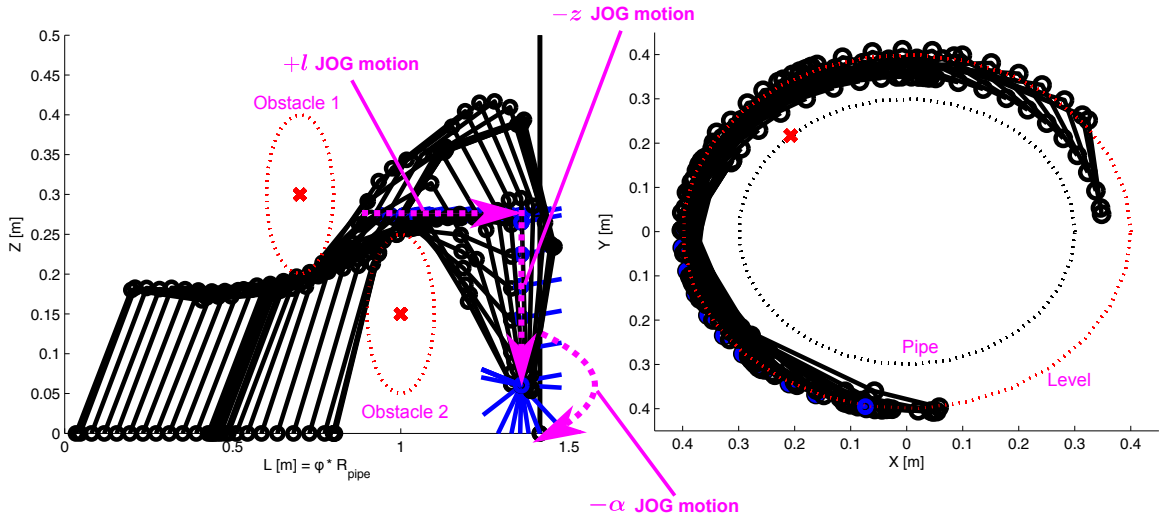


Figure 11: Optimal motion of the proposed manipulator during the coordinated JOG

4 Conclusion

The report deals with the virtual kinematic model of the ROBIN manipulator for NDT of pipe welds in restricted area. The detailed study is devoted to the 6 DoF alternative part of the manipulator where only 5 DoF of the end-effector are controlled (XYZ translation and the last link direction vector), see Chapter 2. The optimal algorithm for minimal joint velocities computation is introduced. The main part of the report was devoted to the virtual model of overall robot (7 DoF conventional and 6 DoF alternative part) and to the design of optimal control algorithm regarding obstacle avoidance. The main features can be summarized as follows:

- An intuitive representation of the trajectory planning (JOG) in the developed view - only four generalized coordinates
- The common obstacles representation in the developed view (surface level, perpendicular pipe inlet envelope) which corresponds to the standard pipeline system and appropriate welds documentation in power plants
- Weighted parametrization of the obstacles avoidance
- Implemented JOG function can be easily modified to optimal joint coordinates computation from prescribed end-effector trajectory (automated weld testing in restricted areas)

All virtual models SW are available on Department of cybernetics at the University of West Bohemia (e-mail: `msvejda@kky.zcu.cz`).

Acknowledgement

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

References

[1] Lukáš Bláha, Vlastimil Šetka, David Tolar, and Petr Barták. HW modules of ADRA-2I conceptual architecture. Technical report, Z'apadočeská univerzita v Plzni, 2018.

[2] 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/`.

[3] Arnold Jáger. Tripod s hydraulickými aktuátory - ověření možnosti řízení pohybu. Technical report, Západočeská univerzita v Plzni, 2017.

[4] Ondřej Ježek. Control electronics of shape-memory alloy wires. Technical report, Západočeská univerzita v Plzni, 2017.

[5] 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`.

[6] Bruno Siciliano. Kinematic control of redundant robot manipulators: A tutorial. *Journal of Intelligent and Robotic Systems*, 3(3):201–212, 1990. URL: `http://dx.doi.org/10.1007/BF00126069`, `doi:10.1007/BF00126069`.

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

[8] 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`.

[9] Martin Švejda. *robotLib* (knihovna předimplementovaních funkcí a funkčních bloků), 2016. URL: `http://home.zcu.cz/~msvejda/PhD_disertace/Algoritmy/robotLib/`.

[10] Martin Švejda. Preliminary architecture design and control algorithm of robin robot. Technical report, University of West Bohemia, Department of Cybernetics, 2017. URL: `http://home.zcu.cz/~msvejda/_publications/2017/3_conventionalCells_control.pdf`.

[11] Martin Švejda. Řízení pohybu prostřednictvím SMA drátu (dráty s tvarovou pamětí). Technical report, Západočeská univerzita, 2017. URL: `http://home.zcu.cz/~msvejda/_publications/2017/5_SMA_rizeni.pdf`.