Elastostatic calibration for cylindrical robot

1 Model

- Since there are rigid links and flexible joints there are 3 parameters which should be identified
  Model scheme:

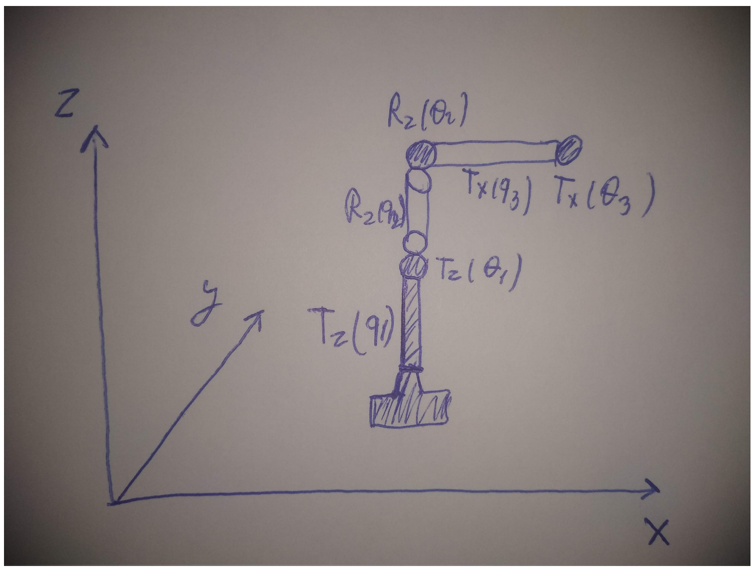

Figure 1: Model scheme

- Direct kinematics for this robot can be written as follows:
  $T = T_z(q_1)T_z(\theta_1)R_z(q_2)Rz(\theta_2)T_x(q_3)T_x(\theta_3))$

- Inverse kinematics:

  - $q_1 = z$
  - $q_2 = \arctan 2(y, x)$
  - $q_3 = \sqrt{x^2 + y^2}$

2 Calibration process

1. Create 30 random configuration. Configuration include 3 joints positions and applied range.

2. Numerically compute jacobians with respect to joints deflections for each configurations.

3. Compute matrix A where each row has a following form:

   $$A_i = \begin{bmatrix} J_{1i} J_{1i}^T W_i & J_{2i} J_{2i}^T W_i & J_{3i} J_{3i}^T W_i \end{bmatrix}$$

   where $J_1 i, J_2 i, J_3 i$ - Jacobians over 1st, 2nd and 3rd joints deflections on i^{th} configurations, W_i - range applied on i^{th} configuration

4. Compute real end-effector deflection on each configuration. k - vector real joints parameters (inverse of joint stiffness) given in problem statement, in reality measured by laser tracker

   $$\Delta t_i = A_i * k$$

5. Compute joint parameters by formula:

   $$k_{est} = (\sum_{i=1}^{m} A_i^T A_i)^{-1} \sum_{i=1}^{m} A_i \Delta t_i$$

3 Error correction

1. Set sequence of points that end-effector should pass and randomly set a load

2. For each point compute inverse kinematics

3. Compute deflection in this point based on estimated parameters

4. Plug to joints position controller inverse kinematics for point which coordinates equal to difference between desired coordinates and deflection.

4 Results analysis

4.1 Without noise in deflection measurement

- As a result of calibration joints stiffness computed precisely since there is no error in deflection measurement.

- Comparison of desired trajectory and obtained by calibrated and non-calibrated robot (here we can see that desired trajectory ideally coincident with one obtained by calibrated robot)

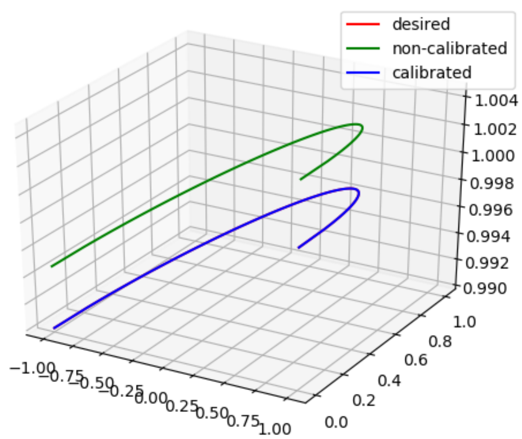

Figure 2: Trajectories plot

- Mean error for non-calibrated robot on x, y and z equal to 4.21, 14.18, 0.66 mm correspondingly.

- For calibrated robot errors on all direction equal to zero

4.2 With noise in deflection measurement

- In reality the laser tracker used for deflection measurement has some noises

- There was also analyzed the calibration result with different noise of laser tracker. Noise treated as normally distributed value with 0 mean
- Calibration process remains the same only difference is that we add noise to Δt_i
- If noise variance equal to 0.01mm:

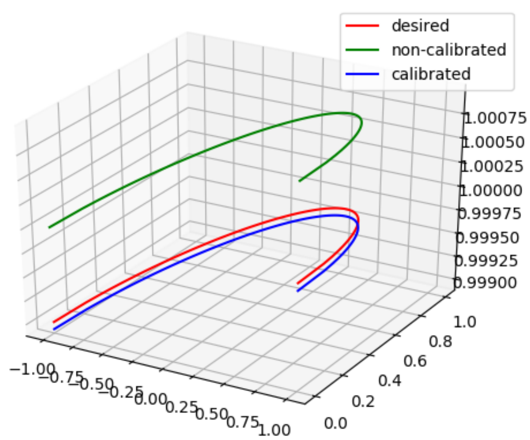

Figure 3: Trajectories plot with 0.01 noise variance

- If noise variance equal to 0.05mm:

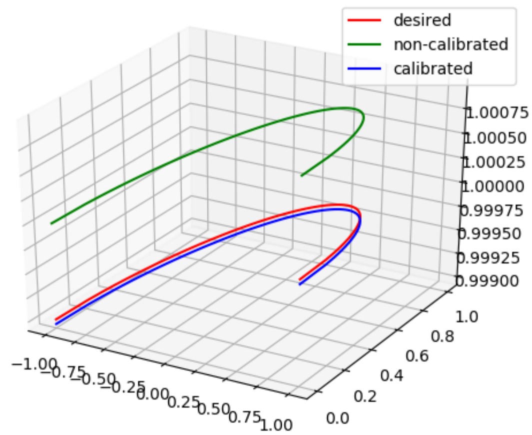

Figure 4: Trajectories plot with 0.05 noise variance

- If noise variance equal to 0.1mm:

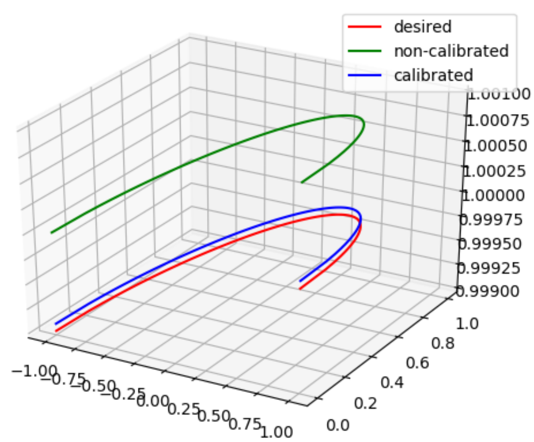

Figure 5: Trajectories plot with 0.1 noise variance

- If noise variance equal to 0.5mm:

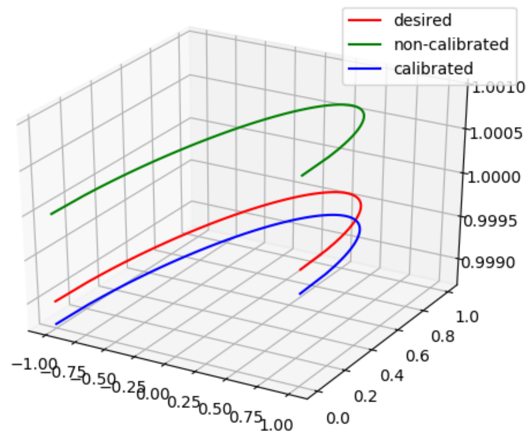

Figure 6: Trajectories plot with 0.5 noise variance

Table with errors dependent on noise (errors taken with precision up to 6 digits after point, comparison was done with range [1000, 1000, 1000, 1000, 1000, 1000]):

Noise variance, m	Mean x error, m	Mean y error, m	Mean z error, m
10^{-5}	0.000012	0.000033	0.000077
$5 * 10^{-5}$	0.000047	0.000032	0.000193
10^{-4}	0.000038	0.000040	0.000308
$5 * 10^{-4}$	0.000662	0.000819	0.000266

5 Github link:

link