

Fundamentels of Robotics

Assignment 6 - Calibration. Innopolis University, Fall 2020

Name: Hany Hamed

Group Number: BS18-Robotics

GitHub Repository: [here](#)

Note: I have changed the report and made the code output more accurate results as I knew that the assignment deadline can be extended till the day of marking the assignment.

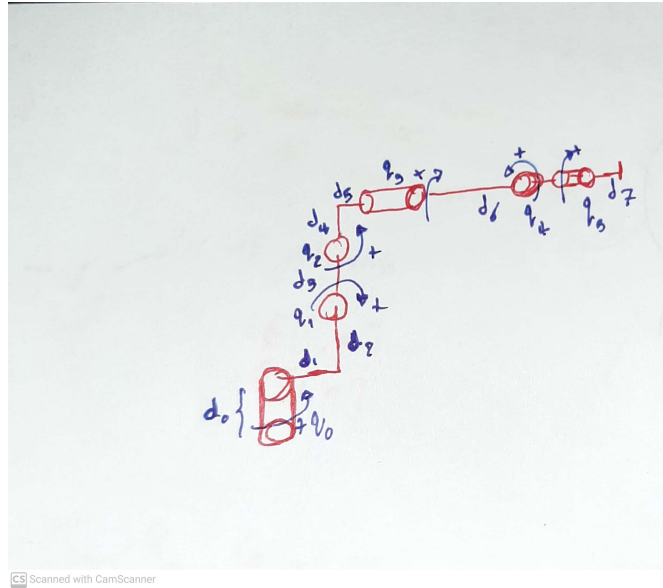
Contents

Fundamentels of Robotics	1
Section 1: Irreducible Model Identification Jacobian	2
Section2: Calibration Algorithm	3
Section 3: Results	4
Section 4: Appendix	6

Section 1: Irreducible Model Identification Jacobian

The robot is a manipulator from Fanuc (R-2000i C/165F) This robot has 6 joints and is 6 DoF manipulator, moreover, it has a spherical wrists.

The model of the robot in zero configuration is as following:



Robot model in zero configuration

d ₀	364 mm
d ₁	312 mm
d ₂	324 mm
d ₃	1075 mm
d ₄	255 mm
d ₅	155.5 mm
d ₆	1124.5 mm
d ₇	215 mm

Note: I am using zero-indexing such that:

The reducible Kinematic Model is as following:

The convention of the notation are from [2, 1]

$$T = [T_x \ T_y \ T_z \ R_x \ R_y \ R_z]_b^T R_z(q_0) T_x(d_1 + Px_0) T_y(Py_0) R_x(\phi_{x0}) R_y(q_1 + \Delta q_1) T_x(Px_1) R_x(\phi_{x1}) R_z(\phi_{z1}) R_y(q_2 + \Delta q_2) T_x(d_5 + Px_3) T_z(d_4 + Pz_2) R_z(\phi_{z2}) R_x(q_3 + \Delta q_3) T_y(Py_3) T_z(Pz_3) R_z(\phi_{z3}) R_y(q_4 + \Delta q_4) T_z(Pz_4) R_z(\phi_{z4}) R_x(q_5) [T_x \ T_y \ T_z]_t$$

$$\pi = [Px_0, Py_0, \phi_{x0}, \Delta q_1, Px_1, \phi_{x1}, \phi_{z1}, \Delta q_2, Px_2, Pz_2, \phi_{z2}, \Delta q_3, Py_3, Pz_3, \phi_{z3}, \Delta q_4, Pz_4, \phi_{z4}]^T$$

Picture for the draft solution with full steps is in the appendix.

For calculating identification jacobian, I have used Numerical derivative method from Assignment 3 ([1], pages 58-61). It is written in the code in details.

Section2: Calibration Algorithm

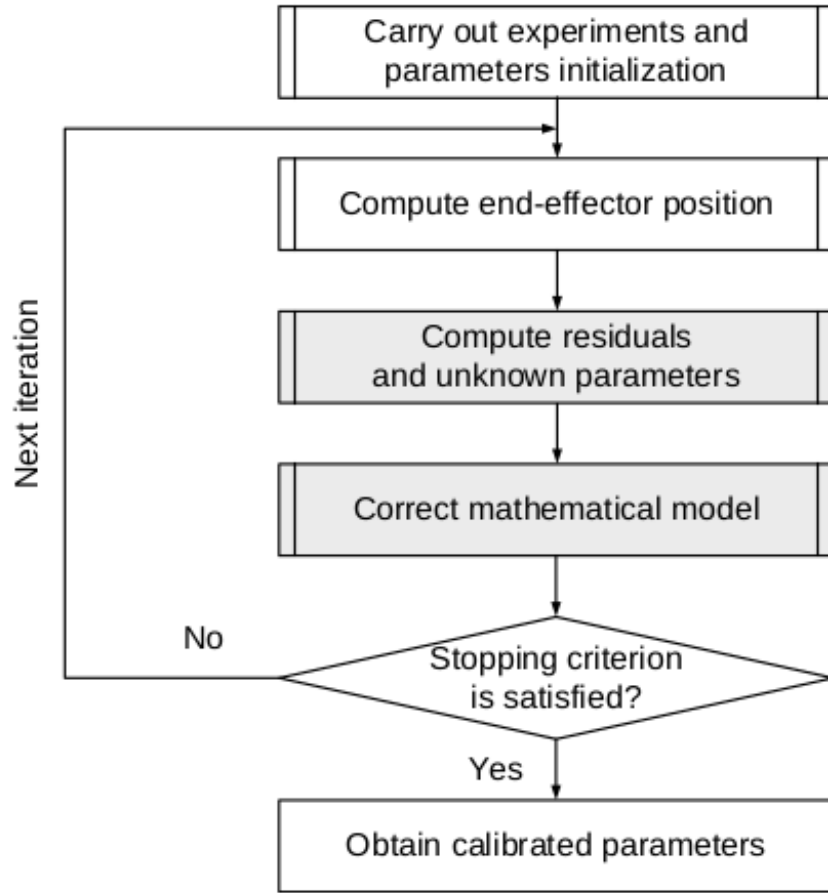
The algorithm is an iterative algorithm that start with the nominal π_0 and improving the results by solving optimization problems. It is separated into 4 parts:

Note: A hand-written draft for the parts and detailed equations are in the Appendix.

- (a) Estimate T_{base} T_{tool} based on knowing π q for each configuration in the experiment.
- (b) Estimate $\Delta\pi$ based on knowing T_{base} T_{tool}
- (c) Update the old values $\pi_{s+1} = \pi_s + \alpha\Delta\pi$
- (d) Termination criteria from the lecture slides: $\sum_{i=1}^m \sum_{j=1}^n ((J_{\pi i}^{j(p)} \cdot \Delta\pi - \Delta P_i^j)^T (J_{\pi i}^{j(p)} \cdot \Delta\pi - \Delta P_i^j))$ if is less than epsilon then terminate. Moreover, I have added a criteria that related to number of iterations with limits to a maximum number of iterations

The full details for the first two steps in ([1], pages 71, 72)

The full algorithm from the lecture slides:



Full algorithm flowchart

Section 3: Results

I have run the code for multiple iterations with some modifications with the initial π_0 and with different step size, I have received the following results after running 100 iterations (It could be improved with more iterations).

(a) π :

$$\begin{bmatrix} -348.098 \\ 263.578 \\ -0.128 \\ -1.545 \\ 1156.484 \\ 3.044 \\ 0.023 \\ 4.064 \\ 370.276 \\ -1964.631 \\ 0.768 \\ -10.457 \\ -72.295 \\ 0.528 \\ 0.181 \\ 4.726 \\ -434.339 \\ -1.578 \end{bmatrix}$$

Such that: $\pi = [Px_0, Py_0, \phi_{x0}, \Delta q_1, Px_1, \phi_{x1}, \phi_{z1}, \Delta q_2, Px_2, Pz_2, \phi_{z2}, \Delta q_3, Py_3, Pz_3, \phi_{z3}, \Delta q_4, Pz_4, \phi_{z4}]^T$
 T_{base} :

$$\begin{bmatrix} 1. & -0.021 & 0.042 & -38.041 \\ 0.021 & 1. & -0.007 & -2.865 \\ -0.042 & 0.007 & 1. & 39.341 \\ 0. & 0. & 0. & 1. \end{bmatrix}$$

T_{tool} :

$$\begin{bmatrix} \begin{bmatrix} 1. & 0. & 0. & 0.177 \\ 0. & 1. & 0. & -7.743 \\ 0. & 0. & 1. & 32.95 \\ 0. & 0. & 0. & 1. \end{bmatrix} \\ \begin{bmatrix} 1. & 0. & 0. & 0.373 \\ 0. & 1. & 0. & 8.444 \\ 0. & 0. & 1. & -6.487 \\ 0. & 0. & 0. & 1. \end{bmatrix} \\ \begin{bmatrix} 1. & 0. & 0. & -5. \\ 0. & 1. & 0. & 35.891 \\ 0. & 0. & 1. & -8.107 \\ 0. & 0. & 0. & 1. \end{bmatrix} \end{bmatrix}$$

RMS Distance Error: 106.22629554605516

Max Distance error (mm): 256.4723020668694

RMS Error for x-coordinate: 48.756062413321224

Max error for x-coordinate (mm): 146.62616119261543

RMS Error for y-coordinate: 73.9786616770403

Max error for y-coordinate (mm): 246.36907091244814

RMS Error for z-coordinate: 58.60059607086086

Max error for z-coordinate (mm): 176.96865366806128

Moreover, I wanted to try a learning based approach by extend this assignment to use CMA-ES as I have seen an implementation for it for system identification for a tensegrity robot and wanted to discover the results and the implementation for this method it in this assignment but did not have time

References

- [1] WU, Y. *Optimal pose selection for the identification of geometric and elastostatic parameters of machining robots*. PhD thesis, 2014.
- [2] WU, Y., KLIMCHIK, A., CARO, S., FURET, B., AND PASHKEVICH, A. Geometric calibration of industrial robots using enhanced partial pose measurements and design of experiments. *Robotics and Computer-Integrated Manufacturing* 35 (2015), 151–168.

Section 4: Appendix

Irreducible Model draft

Accept Modeling

$$T = \underbrace{[T_x \ T_y \ T_z \ R_x \ R_y \ R_z]}_{T_{base}} \cdot \underbrace{R_z(q_0 + \Delta q_0)}_{\substack{q_0 \\ \text{base}}} \cdot \underbrace{[T_x \ T_y \ R_x \ R_y]}_{\substack{q_1 \\ \text{base}}} \cdot \underbrace{R_y(q_1 + \Delta q_1)}_{\substack{q_1 \\ \text{base}}} \cdot \underbrace{[T_x \ T_z \ R_x \ R_z]}_{\substack{q_2 \\ \text{base}}} \cdot \underbrace{R_x(q_2 + \Delta q_2)}_{\substack{q_2 \\ \text{base}}} \cdot \underbrace{[T_y \ T_z \ R_y \ R_z]}_{\substack{q_3 \\ \text{base}}} \cdot \underbrace{R_y(q_3 + \Delta q_3)}_{\substack{q_3 \\ \text{base}}} \cdot \underbrace{[T_x \ T_z \ R_x \ R_z]}_{\substack{q_4 \\ \text{base}}} \cdot \underbrace{R_x(q_4 + \Delta q_4)}_{\substack{q_4 \\ \text{base}}} \cdot [T_x \ T_y \ T_z \ R_x \ R_y \ R_z]_t$$

(2)

$$T_{robot} = R_z(q_0) \cdot [T_x \ T_y \ R_x]_{L_0} \cdot R_y(q_1 + \Delta q_1) \cdot [T_x \ R_x \ R_z]_{L_1} \cdot R_y(q_2 + \Delta q_2) \cdot [T_x \ T_z \ R_z]_{L_2} \cdot R_x(q_3 + \Delta q_3) \cdot [T_y \ T_z \ R_z]_{L_3} \cdot R_y(q_4 + \Delta q_4) \cdot [T_z \ R_z]_{L_4} \cdot R_x(q_5)$$

Diagram showing coordinate frames and transformations:

Base frame (0) to Tool frame (5) via intermediate frames (1, 2, 3, 4). The diagram illustrates the Denavit-Hartenberg (DH) parameters and the corresponding transformation matrices for each joint.

Parameters and transformations:

- Base frame (0) to Frame 1: $R_z(q_0)$
- Frame 1 to Frame 2: $R_y(q_1 + \Delta q_1)$
- Frame 2 to Frame 3: $R_x(q_2 + \Delta q_2)$
- Frame 3 to Frame 4: $R_y(q_3 + \Delta q_3)$
- Frame 4 to Frame 5: $R_x(q_4 + \Delta q_4)$

Final transformation matrix T_{robot} is the product of these transformations and the tool transformation T_{tool} .

Steps for Irreducible Model for the robot

Step 1 in Calibration Algorithm [2]

$$[p_{base}; \varphi_{base}; u_{tool}^1; \dots u_{tool}^n] = \left(\sum_{i=1}^m A_i^T A_i \right)^{-1} \left(\sum_{i=1}^m A_i^T \Delta p_i \right)$$

Step1 main equation

$$\mathbf{u}_{tool}^j = \mathbf{R}_{base} \mathbf{p}_{tool}^j$$

- A matrix calculation

$$\mathbf{A}_l^j = \begin{bmatrix} \mathbf{I} & [\sim \mathbf{p}_{robot}^i]^T & \mathbf{R}_{robot}^i & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{I} & [\sim \mathbf{p}_{robot}^i]^T & \mathbf{0} & \mathbf{R}_{robot}^i & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{I} & [\sim \mathbf{p}_{robot}^i]^T & \mathbf{0} & \mathbf{0} & \dots & \mathbf{R}_{robot}^i \end{bmatrix}$$

A matrix calculation

- Δp_i calculations

$$\Delta \mathbf{p}_i = (\Delta \mathbf{p}_i^1, \dots, \Delta \mathbf{p}_i^n)^T.$$

$$\Delta \mathbf{p}_l^j = \mathbf{p}_l^j - \mathbf{p}_{robot}^i$$

$n=3 \rightarrow$ tool number of references
 $m=2410 \rightarrow$ number of configurations experiments

$i \rightarrow$ experiment number
 $j \rightarrow$ ref. point number

$T_i^j = T_{base} \cdot T_{robot}(q_i, \pi) \cdot T_{tool}^j$
 loc. of ref. points
 $i \in [1, m]$
 $j \in [1, n]$
 manipulator geometry

$P_i^j = P_{base} + R_{base} \cdot P_{robot}(q_i, \pi) + R_{base} \cdot R_{robot}(q_i, \pi) \cdot P_{tool}^j \rightarrow 3mn \text{ scalar equations}$

$R_{base} = [n \ell_{base}] + I$
 $\ell_{base} \rightarrow$ deviations in the base orientation angles $\rightarrow ?$
 $\ell_{base} = \begin{bmatrix} \ell_{\theta_1} \\ \ell_{\theta_2} \\ \ell_{\theta_3} \end{bmatrix}$
 $[n \ell_{base}] = \begin{bmatrix} 0 & -\ell_{\theta_2} & \ell_{\theta_3} \\ \ell_{\theta_1} & 0 & 0 \\ -\ell_{\theta_3} & \ell_{\theta_2} & 0 \end{bmatrix}$

$u_{tool}^j = R_{base} P_{tool}^j \rightarrow P_{tool}^j = R_{base}^{-1} u_{tool}^j$
 $P_i^j = P_{base} + P_{robot}^j - P_{robot}^j [n \ell_{base}] + R_{robot}^j u_{tool}^j \Rightarrow P_i^j = P_{robot}^j + [I [n \ell_{robot}]^T R_{robot}^j] \begin{bmatrix} P_{base} \\ \ell_{base} \\ u_{tool}^j \end{bmatrix}$

apply linear least square using gauss dot

$A_i^j = \begin{bmatrix} \Delta P_i^j \\ \Delta P_i^j \\ \Delta P_i^j \end{bmatrix} = \begin{bmatrix} \Delta P_i^j \\ \Delta P_i^j \\ \Delta P_i^j \end{bmatrix}$
 $A_i^j = \begin{bmatrix} [n \ell_{robot}^j]^T R_{robot}^j & R_{robot}^j & 0 \\ I & [n \ell_{robot}^j]^T R_{robot}^j & R_{robot}^j \\ 0 & 0 & R_{robot}^j \end{bmatrix}$
 $A_i^j = \begin{bmatrix} 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \end{bmatrix}$
 $A_i^j = \begin{bmatrix} 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \end{bmatrix}$
 $A_i^j = \begin{bmatrix} 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \\ 3 \times 3 & 3 \times 3 & 3 \times 3 \end{bmatrix}$

get u_{tool}^j
 get P_{tool}^j
 get R_{base}
 find P_{base}

CS Scanned with CamScanner

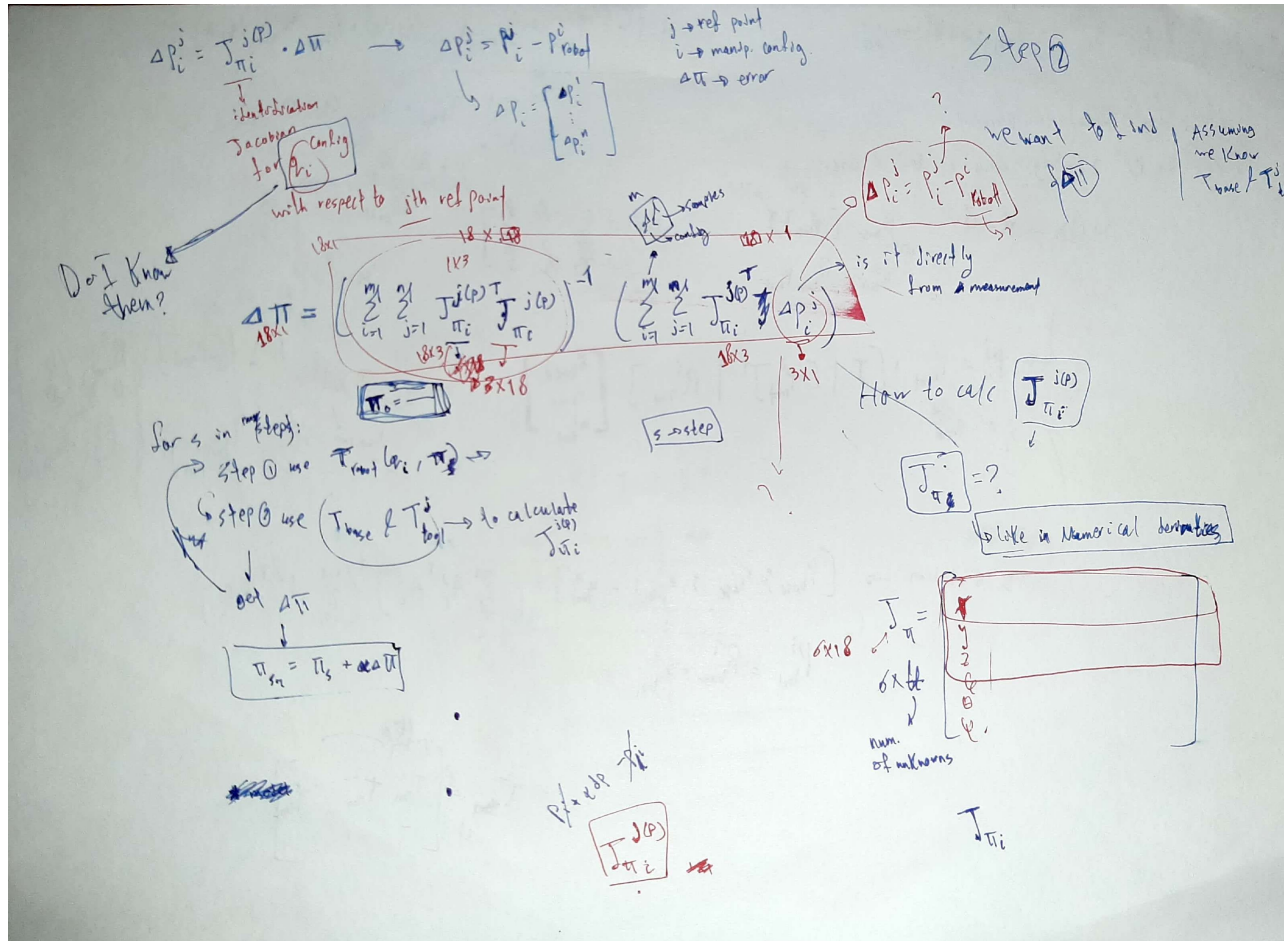
Calibration Algorithm step1 draft

Step 2 in Calibration Algorithm [2]

$$\Delta \pi = \left(\sum_{i=1}^m \sum_{j=1}^n J_{\pi i}^{j(p)T} J_{\pi i}^{j(p)} \right)^{-1} \left(\sum_{i=1}^m \sum_{j=1}^n J_{\pi i}^{j(p)T} \Delta P_i^j \right)$$

Step2 main equation

- Identification jacobian calculation: it is described in details in [1]
- Δp_i calculations: it was mentioned earlier



Calibration Algorithm step2 draft