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**

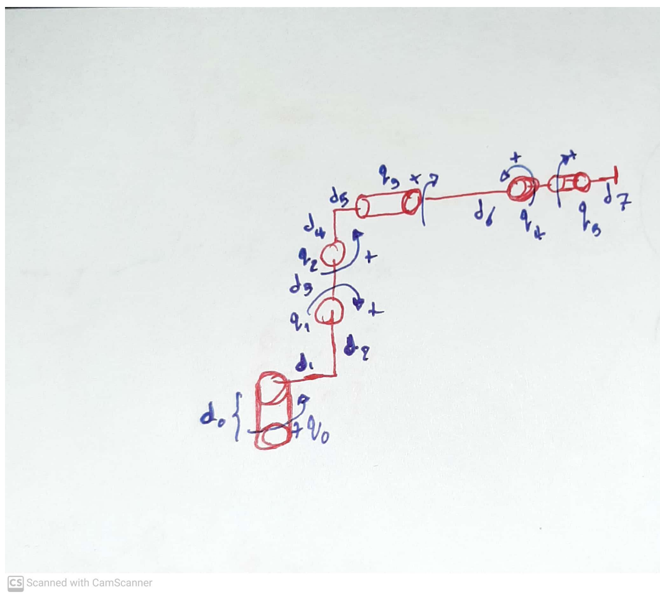
Section 4: Appendix **6**

FoR, Innopolis University, Fall 2020
Author: Hany Hamed

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 writs.

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



Robot model in zero configuration

Note: I am using zero-indexing such that:

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

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 $\pi_o 0$ 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.

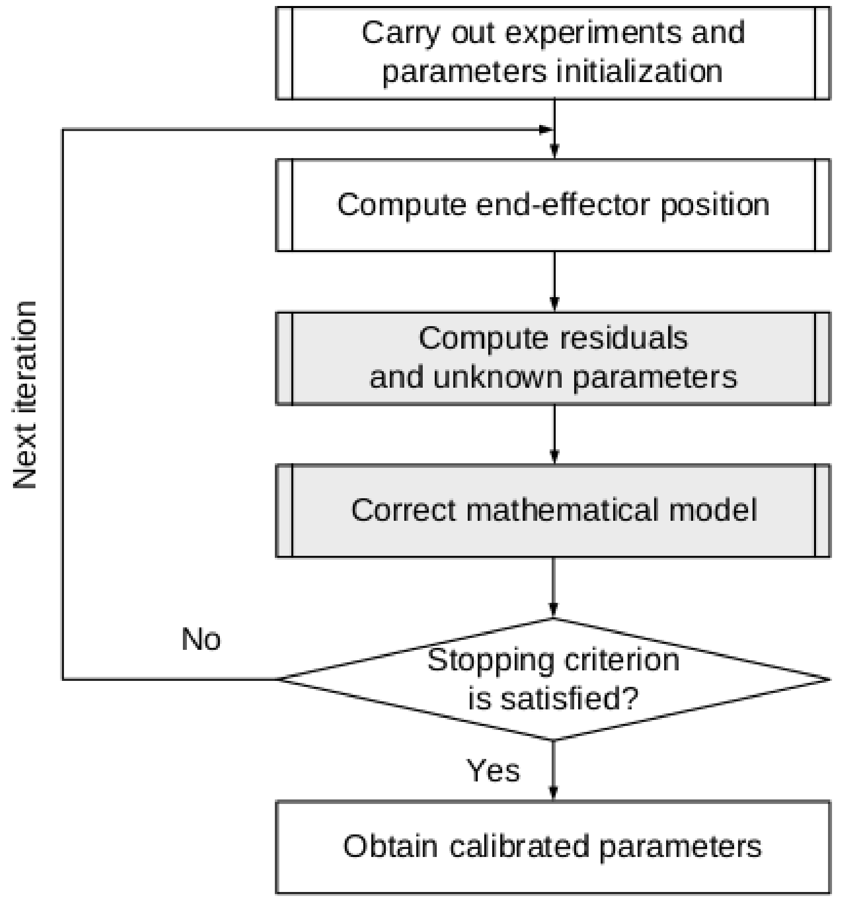
(b) Estimate $\Delta\pi$ based on knowing T_{base} T_{tool}

(c) Update the old values $\pi_{s+1} = \pi_s + \alpha\Delta\pi$

(d) Termination criteria from the lecture slides: $\sum_{i=1}^{m}\sum_{j=1}^{n}((J_{\pi i}^{j(p)}.\Delta\pi - \Delta P_i^j)^T \; (J_{\pi i}^{j(p)}.\Delta\pi - \Delta P_i^j)$ 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

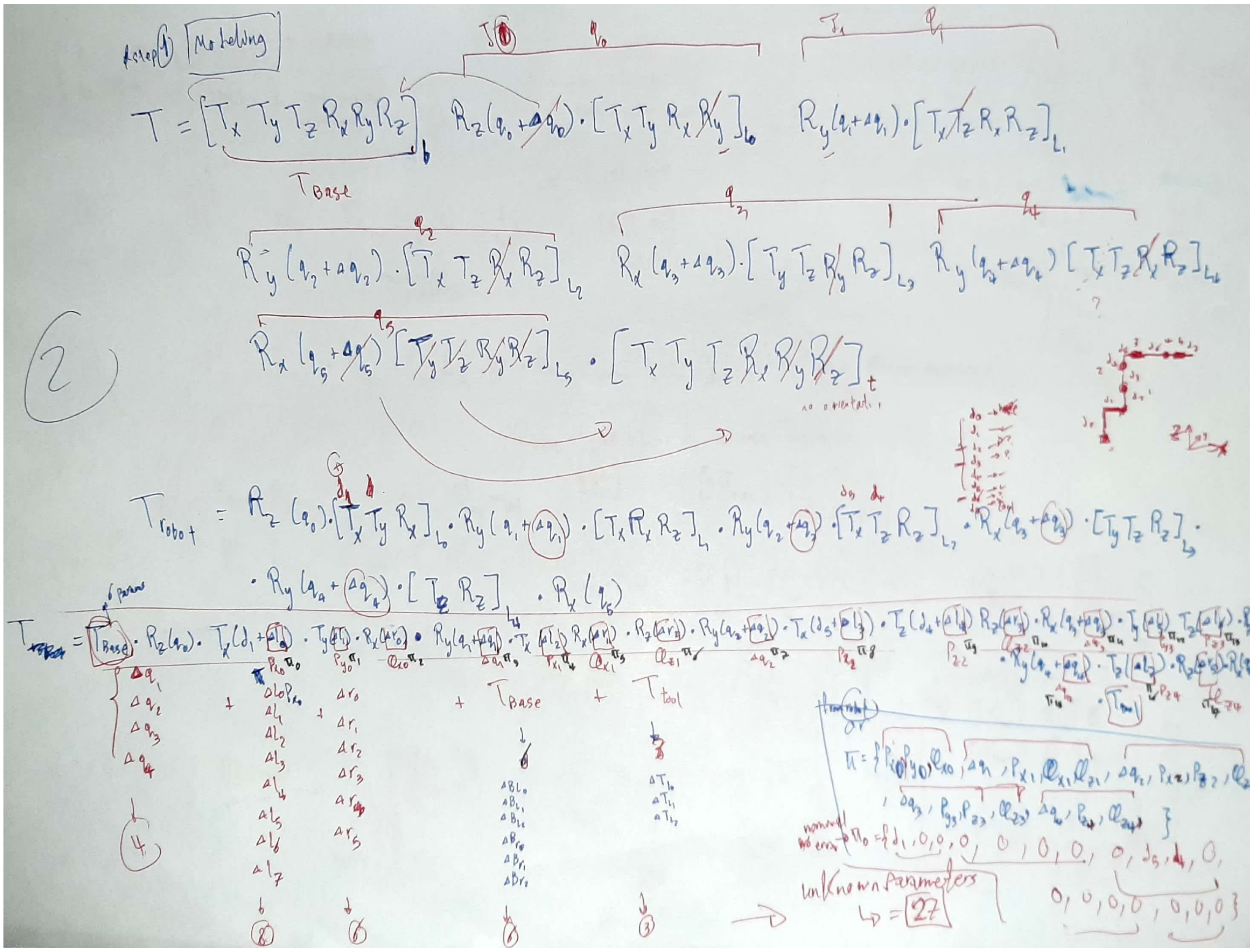
References

[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



Steps for Irreducible Model for the robot

Step 1 in Calibration Algorithm [2]

$$\left[\mathrm{p}_{base};\ \boldsymbol{\varphi}_{base};\ \mathrm{u}^{1}_{tool};\ \ldots\ \mathrm{u}^{n}_{tool}\right] = \left(\sum_{i=1}^{m} \mathrm{A}_i^{j^T} \mathrm{A}_i^{j}\right)^{-1} \left(\sum_{i=1}^{m} \mathrm{A}_i^{j^T} \Delta \mathrm{p}_i\right)$$

Step1 main equation

$$u^j_{tool} = R_{base} p^j_{tool}$$

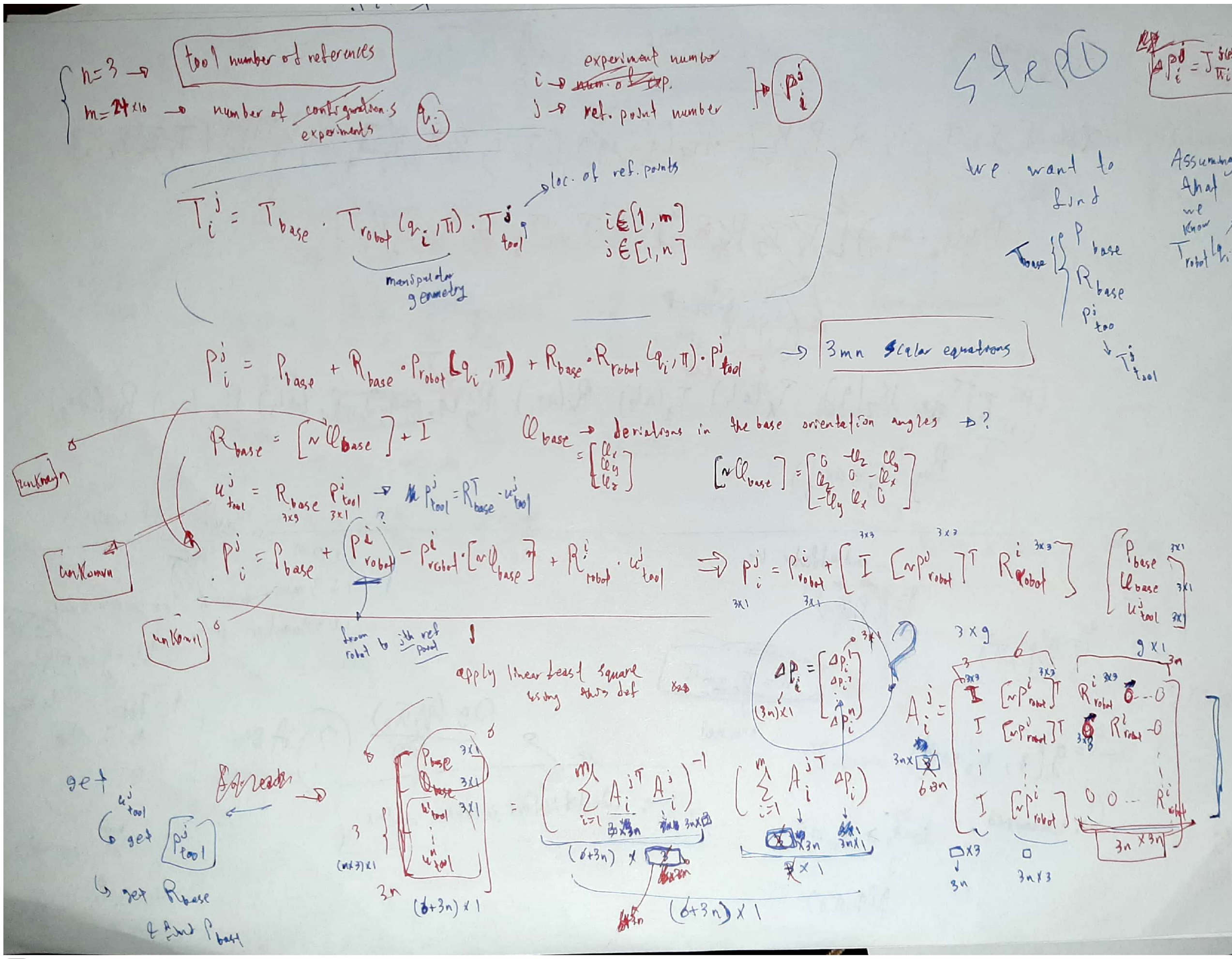
- A matrix calculation

$$A^j_i = \begin{bmatrix} I & \left[\sim p^i_{robot}\right]^T & R^i_{robot} & \mathbf{0} & \dots & \mathbf{0} \\ I & \left[\sim p^i_{robot}\right]^T & \mathbf{0} & R^i_{robot} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ I & \left[\sim p^i_{robot}\right]^T & \mathbf{0} & \mathbf{0} & \dots & R^i_{robot} \end{bmatrix}$$

A matrix calculation

- Δp_i calculations

$$\Delta p_i = (\Delta p^1_i, \dots, \Delta p^n_i)^T.$$
$$\Delta p^j_i = p^j_i - p^i_{robot}$$



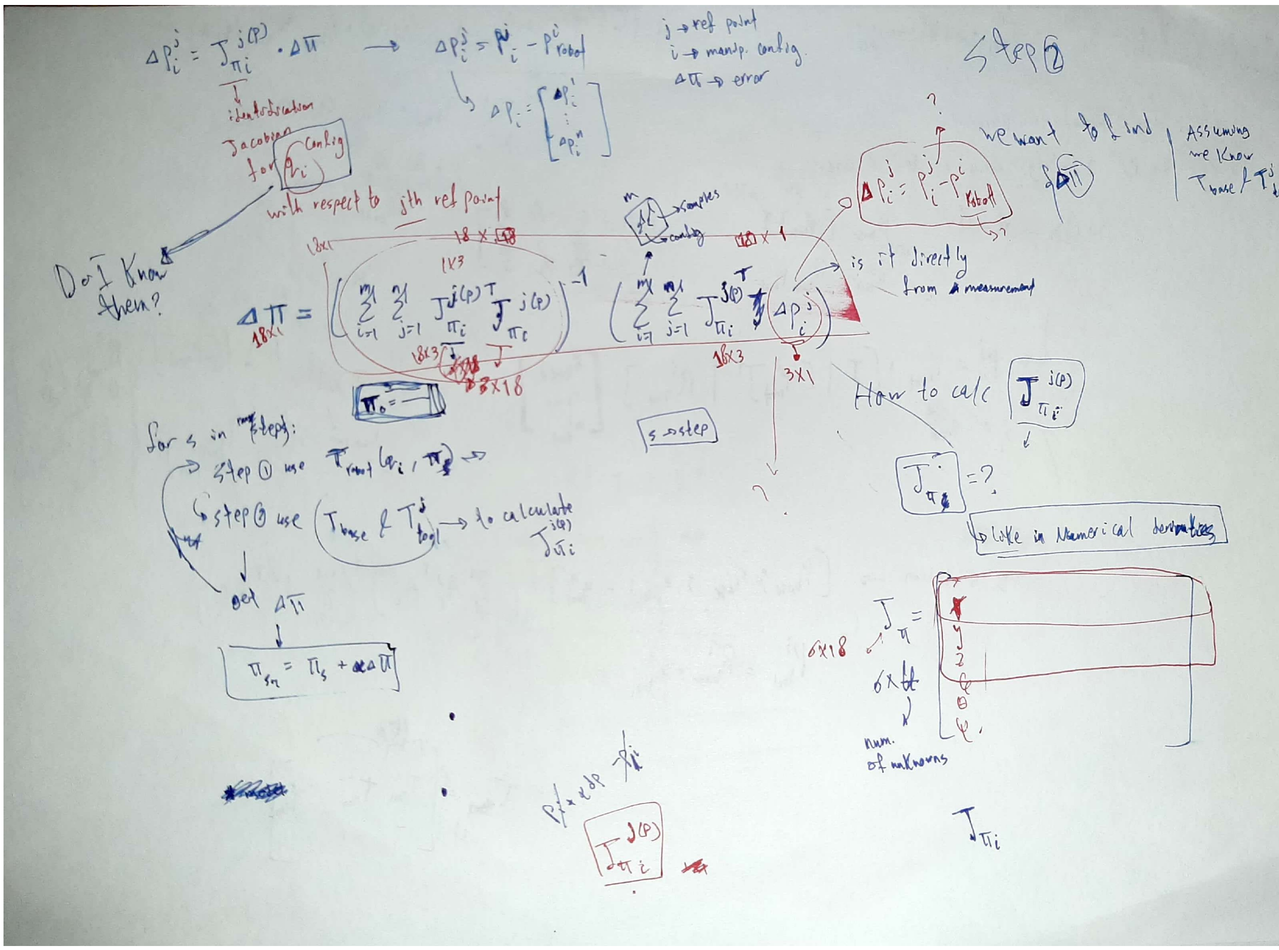
Calibration Algorithm step1 draft

Step 2 in Calibration Algorithm [2]

$$\Delta \boldsymbol{\pi} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{J}_{\pi i}^{j(p)T} \mathrm{J}_{\pi i}^{j(p)} \right)^{-1} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \mathrm{J}_{\pi i}^{j(p)T} \Delta \mathrm{p}_i^j \right)$$

Step2 main equation

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



Calibration Algorithm step2 draft