Fundamentels of Robotics

Assignment 6 - Calibration. Innopolis University, Fall 2020

Name: Hany Hamed
Group Number: BS18-Robotics
GitHub Repository: here

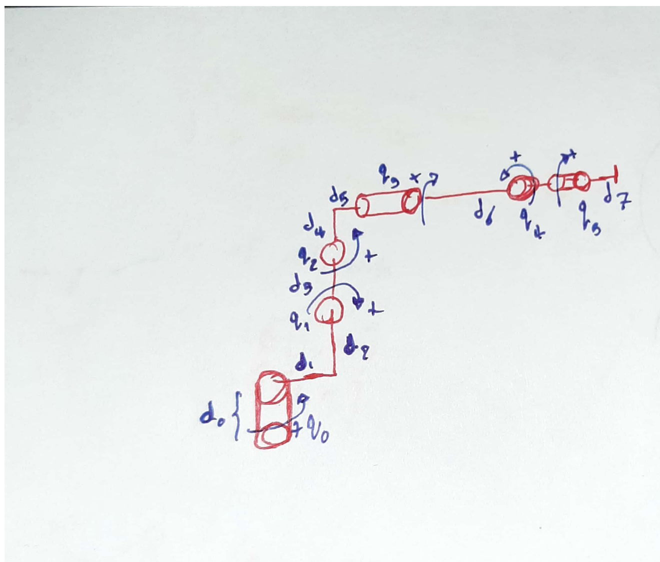
Contents

FoR, Innopolis University, Fall 2020
Author: Hany Hamed

Section 1: Reducible 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:

(a) Estimate T_{base} T_{tool} based on knowing π q for each configuration in the experiment. A draft for the solution is in the following figure.

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

Picture for the draft solution is in the appendix, but for better and illustrative understanding check the code.

(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: TODO: picture from the lecture for the flow

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 (This could be improved by letting the code runs for more iterations

(a) π :
$((-555.794)$
(3309.881)
(0.296)
(-0.344)
(4737.156)

(0.276)
(0.541)
(0.328)
(-1526.099)
(3661.048)
(-0.501)
(-0.185)
(-113.33)
(45.404)
(0.105)
(-0.007)
(-195.059)
(-0.188))

T_{base} :
((1. $-$ 0.511 1.248 $-$ 899.966)
(0.511 1. $-$ 0.259 $-$ 1240.865)
($-$1.248 0.259 1. $-$ 273.891)
(0. 0. 0. 1.))

T_{tool} :

(((1.   0.   0.   -3192.595)
(0.   1.   0.   1529.686)
(0.   0.   1.   -2240.37)
(0.   0.   0.   1.))

((1.   0.   0.   -3208.42)
(0.   1.   0.   1581.136)
(0.   0.   1.   -2234.376)
(0.   0.   0.   1.))

((1.   0.   0.   -3217.408)
(0.   1.   0.   1532.212)
(0.   0.   1.   -2238.681)
(0.   0.   0.   1.)))

RMS Error: 7231.12555725946

Max Distance error (mm): 9094.900265365977

———————————-

RMS Error for x-coordinate: 76.53313656933175

Max error for x-coordinate (mm): 8994.553268508553

———————————-

RMS Error for y-coordinate: 46.630787166701175

Max error for y-coordinate (mm): 4680.645947228523

——————————-

RMS Error for z-coordinate: 41.99144598156676

Max error for z-coordinate (mm): 4913.890037857653

——————————-

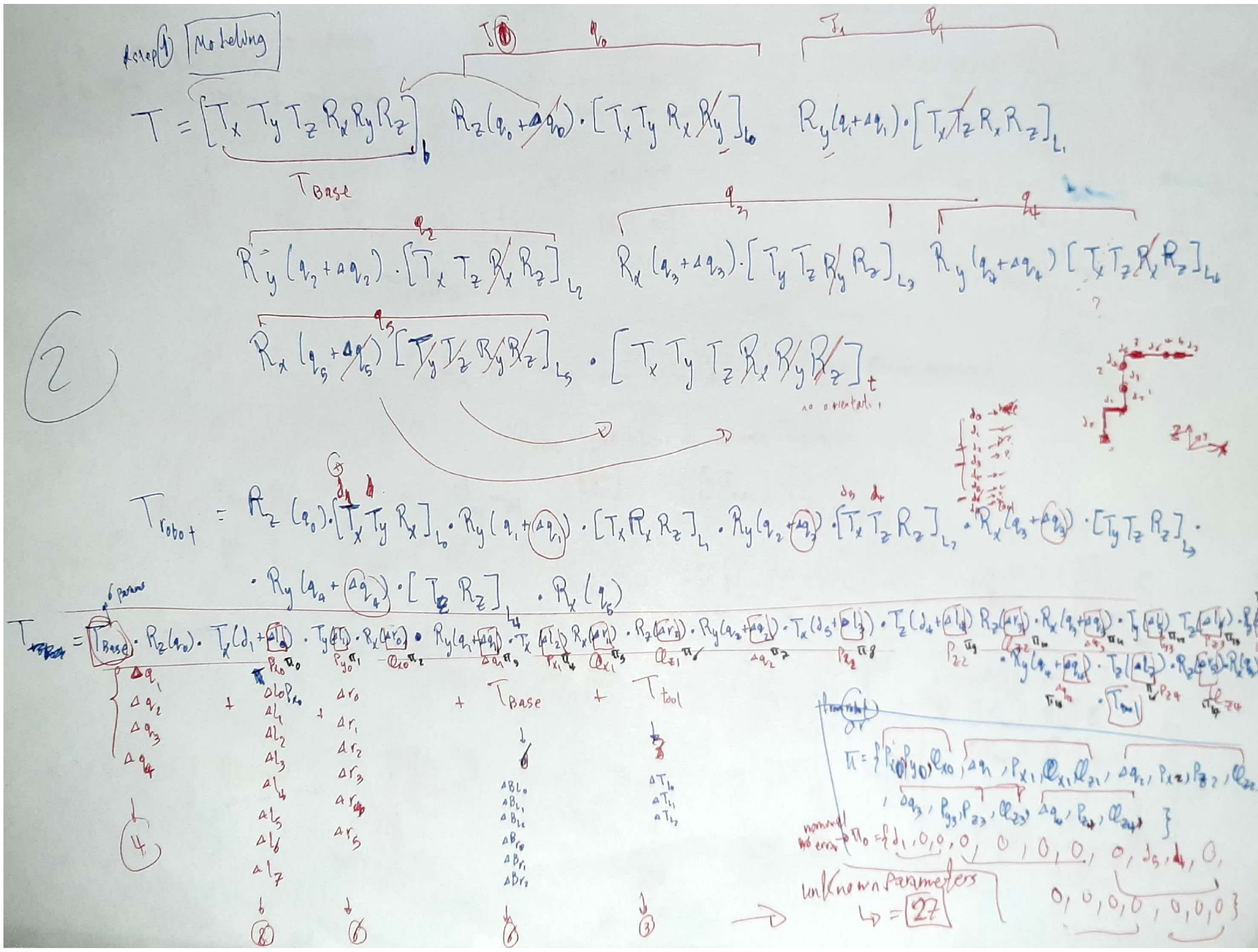
Moreover, I wanted to try a learning based approach by extend this assignment to use CMA-ES as I have seen an implementation for it for system identification for a tensegrity robot and wanted to discover the results and the implementation for this method it in this assignment but did not have time

References

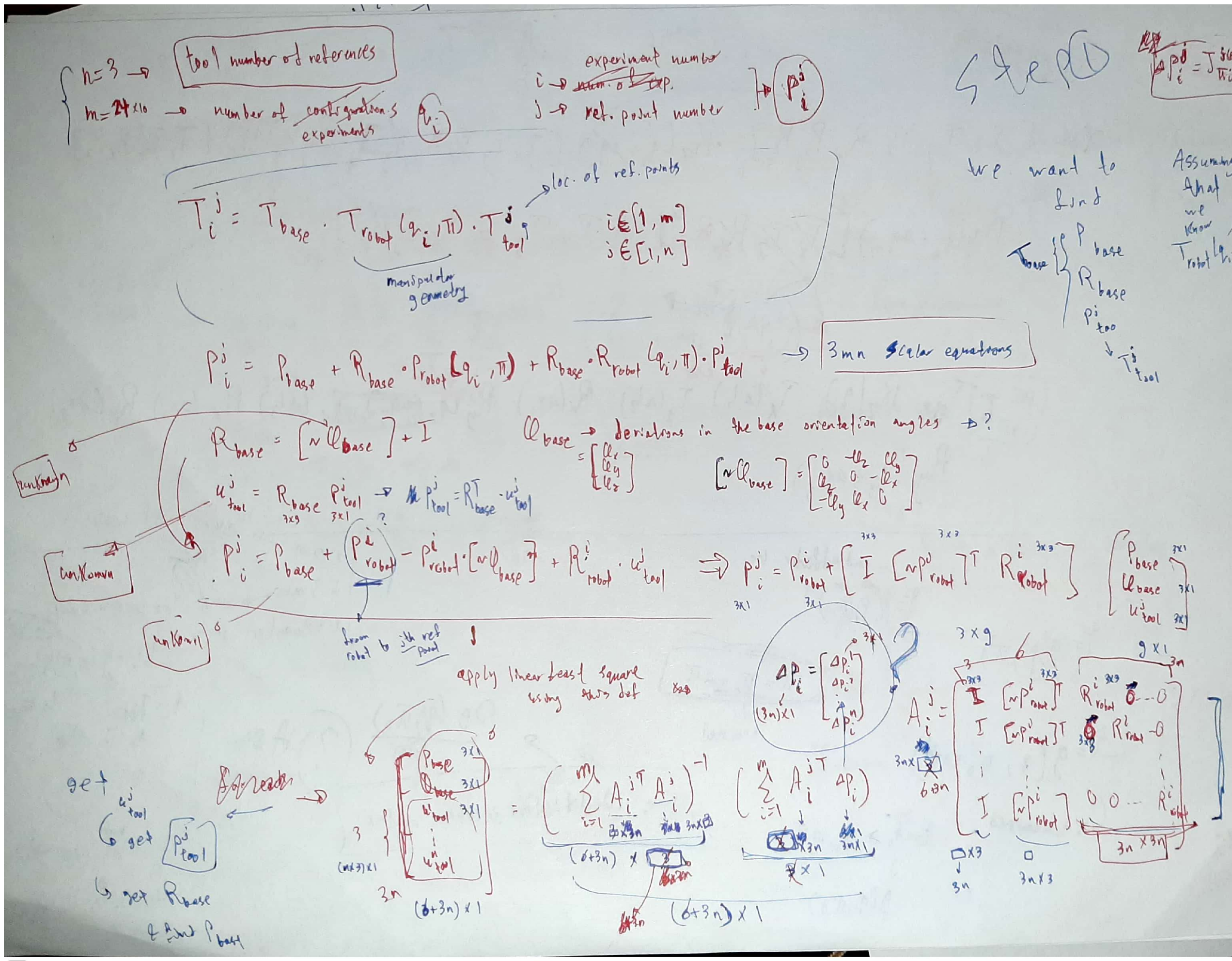
[1] Wu, Y. *Optimal pose selection for the identification of geometric and elastostatic parameters of machining robots.* PhD thesis, 2014.

[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.

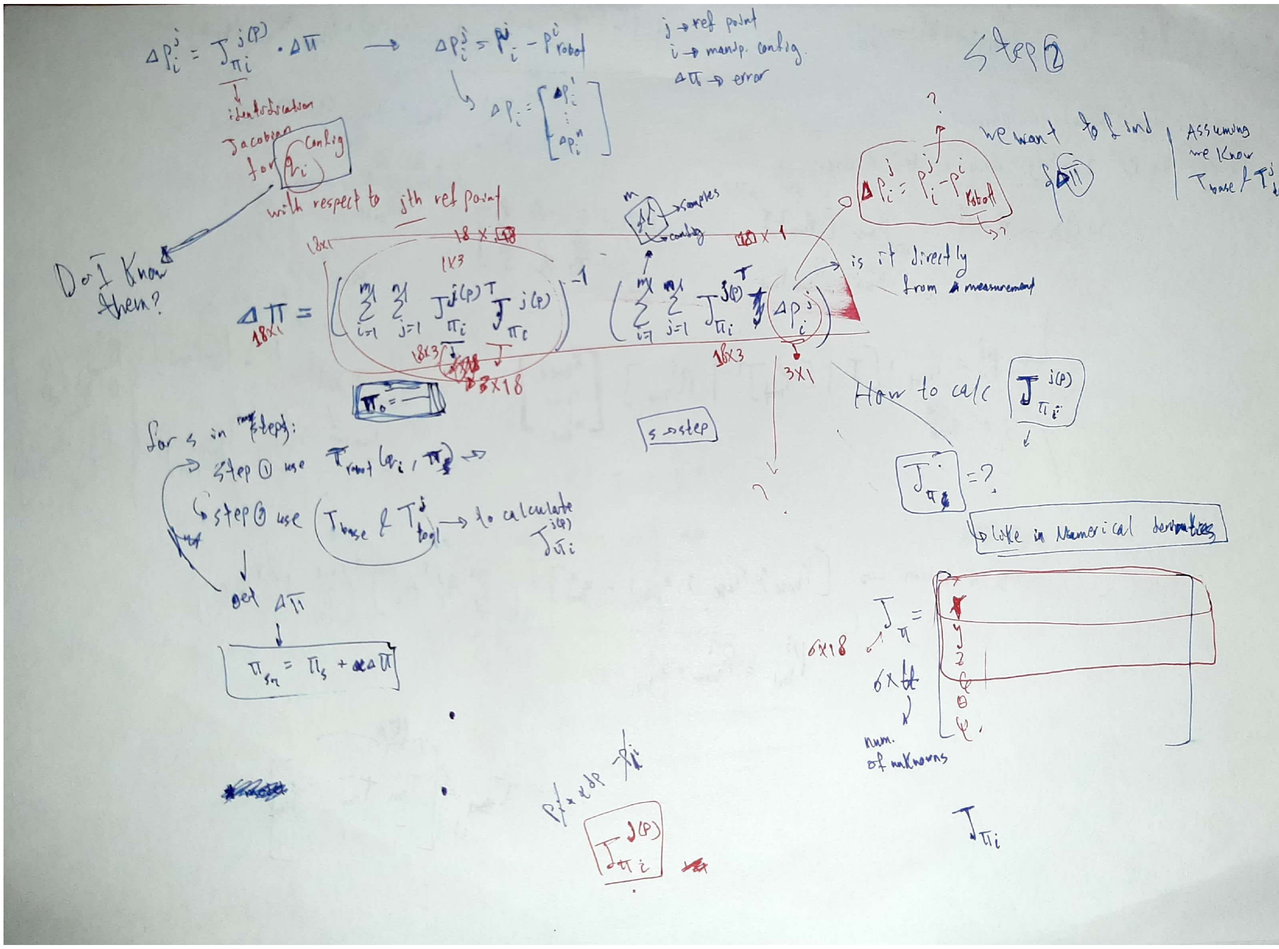
Section 4: Appendix



Steps for reducible Model for the robot



Calibration Algorithm step1 draft



Calibration Algorithm step2 draft