Calibration Pipeline Documentation - Draft 0

CIS II Group 2: Simulation and Kinematic Calibration of the Galen Robot

The calibration pipeline can be ran by running the script *galen_calibration_virtual.m*, which calls in the necessary subfunctions to first perform hand-eye calibration on the robot and the tool attached to it, and then fits and applied a correction polynomial. This document provides the details of each subfunction and explains how they come together in *galen_calibration_virtual.m.*

Main Subfunctions

Solver for the AX = XB Problem

- **File Name:** axxb.m
- **Input:** A and B, two 4x4xN matrices that contain N 4x4 homogeneous transformation matrices
- **Output:** The transformation matrix X

EXPLANATION AND STEPS HERE

Ground Truth Forward Kinematics

- **File Name:** forwardKinematics.m
- **Input:** A 5x1 vector, elements of which are the joint configurations for which we want the forward kinematics
- **Output:** Cartesian position of the end efector, and the tilt and pitch angles
- **Notes:** Provided by Galen Robotics

EXPLANATION AND STEPS HERE

Function for Kinematic Correction

- **File Name:** correct.m
- **Input:** $\mathbf{C}^{data}_{2D}$, and $\mathbf{C}^{cor}_{2D}$, 2D compilations of each C vector measured and corrected respectively.
- **Output:** The matrix **S** that stores the calculated correction distortion coefficient vectors as its rows, and $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$. Also passes on $\mathbf{C}^{cor}_{2D}$.

1. Use the *getCoeffs.m* program inputting $\mathbf{C}^{data\ 2D}$ and $\mathbf{C}^{exp\ 2D}$ to get the coefficient matrix S that stores distortion coefficients calculated in *getCoeffs.m*.
2. Calculate $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$, the vectors containing the minimum and maximum x, y, z coordinates found in our dataset **C**.
3. For each position vector in $\mathbf{C}^{data\ 2D}$, calculate its corrected value by inputting it into the *correctDistortion.m* program alongside **S**, and $\mathbf{q}_{max}$ and $\mathbf{q}_{min}$, the boundaries of the box

we scale the vectors into. Place these inside the 2D matrix $\mathbf{C}^{cor\ 2D}$ which we will print into our output file.

4. Output the calculated **S** and $\mathbf{C}^{cor\ 2D}$, as well as $\mathbf{q}_{max}$ and $\mathbf{q}_{min}$.

Calculating Bernstein Polynomials

- **File Name:** bern.m
- **Input:** The order of the Bernstein polynomial N (which is 5 in our case), the value k = 0, 1, 2, 3, 4, 5 that indicates which coefficient is being calculated, and a scalar v ∈ [0,1].
- **Output:** The Bernstein basis polynomial $B_{N,k}(v)$ defined as [4]:

$$B_{N,k}(v) = \binom{N}{k}(1-v)^{N-k} v^k$$

1. Simply compute the value of the basis polynomial using the equation above.

Calculating Distortion Correction Coefficients

- **File Name:** get_Coeffs.m
- **Input:** Matrices **P** and **Q**, **P** containing the correct (expected) coordinates and Q containing actual measurement coordinates as rows.
- **Output:** The 2D matrix **S** that contains the polynomial coefficients. This matrix is defined as the **C** matrix in the lecture as seen below [4], we called it **S** to avoid confusion with the **C** matrix containing the EM marker position for pivot calibration.

$$\begin{bmatrix} & \vdots & \\ F_{000}(\vec{\mathbf{u}}_s) & \cdots & F_{555}(\vec{\mathbf{u}}_s) \\ & \vdots & \end{bmatrix} \begin{bmatrix} c^x_{000} & c^y_{000} & c^z_{000} \\ \vdots & \vdots & \vdots \\ c^x_{555} & c^y_{555} & c^z_{555} \end{bmatrix} \cong \begin{bmatrix} & \vdots & \\ p^x_s & p^y_s & p^z_s \\ & \vdots & \end{bmatrix}$$

1. From **Q**, calculate $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$ (could also be inputted into the function) from the maximum and minimum x, y, and z values found in **Q**.
2. Scale every **q** (every row of **Q**) and every **p** (every row of **P**) to box to be used with Bernstein polynomials, using *ScaleToBox.m.* Store these scaled **p** values in $\mathbf{P}_{normalized.}$
3. Calculate the Bernstein polynomials B using *bern.m* for each **q** to calculate $F_{i,j,k}$ for each $\mathbf{u}_s$ and construct the **F** matrix.

$$F_{i,j,k} = B_{5,i}(u_x)B_{5,j}(u_y)B_{5,k}(u_z)$$

4. Calculate the matrix $\mathbf{S} = \mathbf{F}^{-1}\mathbf{P}_{normalized}$.

Correcting Distortion Using the Coefficients

- **File Name:** correctDistortion.m
- **Input:** The 3D vector **q** that is to be corrected, $\mathbf{q}_{max}$ and $\mathbf{q}_{min}$ as (will be) calculated in Problem 2, and the matrix **S**, containing the correction coefficients..
- **Output:** **p**, the dewarped (corrected) values of the measured vector **q.**

1. Use *ScaleToBox.m* to scale the input **q** into **u**.
2. For each i, j, k value, access the coefficient vector $\mathbf{c}_{i,j,k}$ from the matrix **S** (as the vectors **c** constitute the rows of matrix **S)**, and multiply it by the Bernstein polynomials as

$$\mathbf{c}_{i,j,k} B_{5,i}(u_x) B_{5,j}(u_y) B_{5,k}(u_z)$$

3. Sum the coefficients vector multiplied by the polynomials across all i,j,k to obtain **p**, the vector of corrected values of **q**, as

$$\mathbf{p} = \sum_{i=0}^{5} \sum_{j=0}^{5} \sum_{k=0}^{5} \mathbf{c}_{i,j,k} B_{5,i}(u_x) B_{5,j}(u_y) B_{5,k}(u_z)$$

Helper Functions

Functions used for “scaling to box”, converting between vectors and matrices, construct rotations from angles, etc.

Scaling To Box

- **File Name:** ScaleToBox.m
- **Input:** A 5D vector of DOFs **q** to be scaled, and $\mathbf{q}_{max}$, $\mathbf{q}_{min}$, the vectors constructed from the maximum and minimum individual values of x, y, z, theta and psi found in our entire data set **C**
- **Output:** A scaled vector **u,** each of its elements u_x, u_y, u_z ,u_t ,$u_p \in [0,1]$.

1. Calculate each of the elements u_x, u_y, u_z in the following manner:

$$u_x = \frac{q_x - q_{min\,x}}{q_{max\,x} - q_{min\,x}}$$

2. Combine each element into the vector $\mathbf{u} = [u_x\ u_y\ u_z\ u_t\ u_p]'$.

Skew Matrix to Vector Converter

- **File Name:** skew2vector.m
- **Input:** A 3x3 skew matrix.
- **Output:** A 3x1 vector containing the non-zero elements of the skew matrix.

rotx

XXX

roty

XXX