Mathematics of manipulator CSYS <-> external CSYS calibration

v1.0: 2022 - 08 - 12

Note: I will use "position" and "coordinate(s)" interchangeably. Both instances refer to a set of values that define a spatial location relative to a coordinate system. For instance $(10, 20)$ *is the position or coordinates of some point in 2D space.*

Introduction

With the Autoinjector, our goal is to robotically target locations in 3D space for microinjection. To do this, we need to accurately direct the micropipette to these 3D locations and deliver our injection payload. However, this is not a trivial task. When we look at tissue through the microscope and camera, we are visualizing the tissue in a coordinate system (CSYS) defined by the camera and the microscope focus position. For instance, the location of a cell in the coordinate system might be $x = 350$ pixels, $y = 900$ pixels, and $z = 1$ mm. But when we are targetting this cell for microinjection, we need to "tell" the micromanipulator where to go relative to its own coordinate system. The same cell with a position of $(350, 900, 1)$ in the microscope/camera CSYS might have a position of $x = 14000$ nm, $y = 9560$ nm, $z = 18775$ nm, and $d = 16990$ nm. In this case, d is the manipulator's injection axis.

So now we have a single location in 3D space defined by two distinct coordinate systems: $p_{external} = (350, 900, 1)$ is the postion of the location relative to the external coordinate system (the microscope and camera), and $p_{manipulator} = (14000, 9560, 18775, 16990)$ is the position of the location relative to the manipulator's coordinate system. Now the question is, "Given the location of the micropipette tip as a manipulator postion what is its position relative to the external CSYS? Likewise, what is the manipulator position given a external position?". We can answer these questions by computing a calibration.

The goal of calibration is to construct a map: a map that tells the robot what the position of the micropipette tip is in the external coordinate system when we know it the manipulator's position. We can then use the "inverse" of this map to tell the robot what its manipulator position should be to reach a location defined by an external position.

Calibration model

As mentioned we want to define a map, M, that transforms between an manipulator position at some time instance to an external position at the same time

instance. Formally:

$$M : p_m(k+1) \rightarrow p_{ex}(k+1)$$

where p is the position, subscripts $_{ex}$ and $_m$ denote the external and manipulator coordinate systems, and $(k+1)$ defines the time instance. In the case of our robotic system where our microscope has a camera and motorized focus mechanism, the external position has 3 entries (i.e. $p_{ex} \in \mathbb{R}^3$). Conversely, our Sensapex micromanipulator has 4 axes (because it has diagonal injection axis), so the manipulator position has 4 entries (i.e. $(p_m \in \mathbb{R}^4)$).

$$p_{ex} = \begin{bmatrix} x_{ex} \\ y_{ex} \\ z_{ex} \end{bmatrix}$$

$$p_m = \begin{bmatrix} x_m \\ y_m \\ z_m \\ d_m \end{bmatrix}$$

We will assume a very simple discrete model for our robotic system. We assume that the external position of the micropipette tip at some time instance is the tip's previous external position plus the change in the tip's external position.

$$p_{ex}(k+1) = p_{ex}(k) + (p_{ex}(k+1) - p_{ex}(k))$$

$$p_{ex}(k+1) = p_{ex}(k) + \Delta p_{ex}(k)$$

However, the system has no explicit knowledge of change in the tip's external position. For instance, the user may observe on the video display that the tip has moved from $x_{ex}(k) = 350$ pixels to $x_{ex}(k+1) = 600$ pixels, but this information isn't know to the computer. The computer has no measurement system to measure the tip position in the camera image (yet). But the system always has access to the manipulator's position, so we want to redefine $\Delta p_{ex}(k)$ in terms of the the change in manipulators positions, $\Delta p_m(k) = p_m(k+1) - p_m(k)$.

We will define a transformation, $\hat{T}$, that transforms between a change in manipulator position to a change in external position. Specifically:

$$\hat{T} : \Delta p_m(k) \rightarrow \Delta p_{ex}(k)$$

$$\Delta p_{ex}(k) = \hat{T} \times \Delta p_m(k)$$

$$\Delta p_{ex}(k) = \hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4} \times \Delta p_m(k)$$

We can assume that $\hat{T}$ is a composition of two separate transformations. The first transformation, $\hat{T}_{m_3 \leftarrow m_4}$, transforms the manipulator's displacement along its d axis to projected displacements along the manipulators x, y, and z axes. The second transformation, $\hat{T}_{ex \leftarrow m_3}$, transforms displacements along the manipulators x, y, and z axes to the external x, y, and z axes.

This composition allows an intutive explanation of how the manipulator's displacements are translated to external displacements. First we account for how moving the manipulator d axis is equivalent to moving its other axes. For instance, if we assume the d axis is coplanar with the manipulator's x - z plane, then displacing the manipulator d axis is equivalent to displacing the manipulator's x and z axes by some amount. After this, then we can account for how displacing the manipulator's x, y, and z axes relates to displacements in the external x, y, and z axes.

Therefore, we can update our initial equation for the external position as

$$p_{ex}(k+1) = p_{ex}(k) + \hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4} \times \Delta p_m(k)$$

When we want to know the external position of the micropipette tip, we simply compute the difference between our manipulator positions (which is possible because we always have access to its position), we multiply it by our transformation matrices, and finally add this to our intial external position. Simple! To do this though, we need to know the equation parameters: namely the initial external position, $p_{ex}(k)$, and the transformation matrices, $\hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4}$.

Defining model parameters

For our intial system, we are going to make some assumptions that will vastly simplify our parameter estimation for $p_{ex}(k)$, $\hat{T}_{ex \leftarrow m_3}$, and $\hat{T}_{m_3 \leftarrow m_4}$. These assumptions will simplify the structure of the transformation matrices and reduce the number of parameters we will

Assumptions

1. The manipulator's d- and y-axes are perpendicular
   - ($\Delta d_m \perp \Delta y_m$)
   - Explanation: This implies that some displacement in the manipulator's d-axis could be replicated by a combination of displacements in the x- and d-axes. Aka. Moving the injection axis "looks" like a combination of going forwards/backwards and up/down.
2. The manipulator's x-, y-, and z- axes are perpendicular to each other.
   - ($\Delta z_m \perp \Delta x_m$ and $\Delta z_m \perp \Delta y_m$ and $\Delta x_m \perp \Delta y_m$)

- Explanation: Maybe this assumption should actually be that none of the axes (not including d) are a linear combination of each other? This means that with moving just x, y, and z axes allows you to acheive any location in 3D space.

3. The angle between the manipulator's x-axis and d-axis is known. (Wtih assumptions (1) and (2), this also defines the angle between the manipulator's z-axis and d-axis)
   - $cos(\theta) = \frac{\Delta x_m}{\Delta d_m}$
   - $sin(\theta) = \frac{\Delta z_m}{\Delta d_m}$
   - Explanation: These equalities are predicated on perpendicuarity. If we modify (2) to be linearly independent (rather than perpendicular) then these equalties down hold. Anyways, this means that by knowing the angle, we can transform the d-axis displacement into an equivalant combination of x- and z-displacements.
4. The external x-, y-, and z- axes are all perpendicular to each other.
   - ($\Delta z_{ex} \perp \Delta x_{ex}$ and $\Delta z_{ex} \perp \Delta y_{ex}$ and $\Delta x_{ex} \perp \Delta y_{ex}$)
   - Explanation: The camera plane is square and is perpendicular to the focus axis.
5. The manipulator's z-axis is parallel to the external z-axis. (With assumption (4) this also implies that the manipulator x-y plane is parallel with the external x-y plane)
   - ($\Delta z_m \parallel \Delta z_{ex} \rightarrow \Delta z_{ex} = \alpha_z \Delta z_m$)
   - Explanation: The camera plane is square and is perpendicular to the focus axis.
6. The manipulator z-axis units are nanometers and the external z-axis units are micrometers.
   - ($\alpha_z = \pm \frac{1}{1000} \frac{\mu\text{m}}{\text{nm}}$)
   - Explanation: The scaling between the z-axes is already known (doesn't need to be determined).

Structure of matrices

With these assumptions we can now define the structure of our transformation matrices. We will uses the $\prime$ notation to define the pseudo-axis displacements (the projected axis displacements as a result of a d-axis change).

Recall that the $\hat{T}_{m_3 \leftarrow m_4}$ matrix transforms the manipulator's displacement along its d axis to projected displacements along the manipulators x, y, and z axes.

With assumptions (2) and (3) we have the feollowing equation for a displacement along the manipulator x-axis.

$$\Delta x_m\prime = 1 \times \Delta x_m + 0 \times \Delta y_m + 0 \times \Delta z_m + \cos\theta \times \Delta d_m$$

With assumptions (1) and (2) we have the following equation for a displacement along the manipulator y-axis.

$$\Delta y_m\prime = 0 \times \Delta x_m + 1 \times \Delta y_m + 0 \times \Delta z_m + 0 \times \Delta d_m$$

With assumptions (2) and (3) we have the feollowing equation for a displacement along the manipulator x-axis.

$$\Delta z_m\prime = 0 \times \Delta x_m + 0 \times \Delta y_m + 1 \times \Delta z_m + \sin\theta \times \Delta d_m$$

Putting these equations into matrix form, we define the structure of $\hat{T}_{m_3 \leftarrow m_4}$ matrix.

$$\begin{bmatrix} \Delta x_m\prime \\ \Delta y_m\prime \\ \Delta z_m\prime \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cos(\theta) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sin(\theta) \end{bmatrix} \begin{bmatrix} \Delta x_m \\ \Delta y_m \\ \Delta z_m \\ \Delta d_m \end{bmatrix}$$

$$\hat{T}_{m_3 \leftarrow m_4} = \begin{bmatrix} 1 & 0 & 0 & \cos(\theta) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sin(\theta) \end{bmatrix}$$

Recall that the $\hat{T}_{ex \leftarrow m_3}$ matrix transforms displacements along the manipulators x, y, and z axes to the external x, y, and z axes.

With assumptions (2) and (3) we have the feollowing equation for a displacement along the manipulator x-axis.

$$\Delta x_{ex} = \alpha_{1,1} \times \Delta x_m\prime + \alpha_{1,2} \times \Delta y_m\prime + 0 \times \Delta z_m\prime$$

With assumptions (1) and (2) we have the following equation for a displacement along the manipulator y-axis.

$$\Delta y_{ex} = \alpha_{2,1} \times \Delta x_m\prime + \alpha_{2,2} \times \Delta y_m\prime + 0 \times \Delta z_m\prime$$

With assumptions (2) and (3) we have the feollowing equation for a displacement along the manipulator x-axis.

$$\Delta z_{ex} = 0 \times \Delta x_m\prime + 0 \times \Delta y_m\prime \pm \alpha_z \times \Delta z_m\prime$$

Putting these equations into matrix form, we define the structure of the $\hat{T}_{m_3 \leftarrow m_4}$ matrix.

$$\begin{bmatrix} \Delta x_{ex} \\ \Delta y_{ex} \\ \Delta z_{ex} \end{bmatrix} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 \\ \alpha_{2,1} & \alpha_{2,2} & 0 \\ 0 & 0 & \pm\alpha_z \end{bmatrix} \begin{bmatrix} \Delta x_m\prime \\ \Delta y_m\prime \\ \Delta z_m\prime \end{bmatrix}$$

$$\hat{T}_{ex\leftarrow m_3} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 \\ \alpha_{2,1} & \alpha_{2,2} & 0 \\ 0 & 0 & \pm\alpha_z \end{bmatrix}$$

Estimating model parameters

During the calibration process, the user "clicks" on the in focus pipette tip in the video display which causes the system to not the clicked pixel location, the focus height location, and the manipulator location. This allows us to associate an external position (pixel location and focus height location) with the manipulator position. This gives us two matrices: P_{ex} which is a 'list' of external positions (p_{ex}) and P_m which is a 'list' of manipulator positions (p_m). With these matrices we are able to completely define all the parameters of our model.

$$P_{ex} = \begin{bmatrix} p_{ex,1}^T \\ p_{ex,2}^T \\ \vdots \\ p_{ex,n}^T \end{bmatrix} = \begin{bmatrix} x_{ex,1} & y_{ex,1} & z_{ex,1} \\ x_{ex,2} & y_{ex,2} & z_{ex,2} \\ \vdots & & \\ x_{ex,n} & y_{ex,n} & z_{ex,n} \end{bmatrix}$$

$$P_m = \begin{bmatrix} p_{m,1}^T \\ p_{m,2}^T \\ \vdots \\ p_{m,n}^T \end{bmatrix} = \begin{bmatrix} x_{m,1} & y_{m,1} & z_{m,1} & d_{m,1} \\ x_{m,2} & y_{m,2} & z_{m,2} & d_{m,2} \\ \vdots & & & \\ x_{m,n} & y_{m,n} & z_{m,n} & d_{m,n} \end{bmatrix}$$

$\hat{T}_{m_3\leftarrow m_4}$ **parameters** For $\hat{T}_{m_3\leftarrow m_4}$, this matrix is already completely defined. Above, we showed the structure of this matrix and it can be seen that there is a single variable in this matrix (θ). But we said in assumption (3) that we know this angle θ, so no additional work is needed to define this matrix. The manipulator has a function that returns the angle, so all we need to do is 'ask' the manipulator.

$\hat{T}_{ex\leftarrow m_3}$ **parameters** For $\hat{T}_{ex\leftarrow m_3}$, we have 5 variables shown above ($\alpha_{1,1}$, $\alpha_{1,2}$, $\alpha_{2,1}$, $\alpha_{2,2}$, and α_z). We can eliminate the need to esitmate α_z because we assumed its value in assumptions (5) and (6). (We still need to define the directionality between the axes (+/-) but we leave that to be set by the user. This directionality can be saved and reloaded, so it doesn't need to be set every time.) This leaves four values to be determined ($\alpha_{1,1}$, $\alpha_{1,2}$, $\alpha_{2,1}$, and $\alpha_{2,2}$). We will determine these values by a least squares regression.

For a stationary manipulator d-axis we have the following relationship (same as the equations above that defined the $\hat{T}_{ex\leftarrow m_3}$ matrix while omitting 0 entries):

$$\Delta x_{ex} = \alpha_{1,1} \times \Delta x_{m'} + \alpha_{1,2} \times \Delta y_{m'}$$

$$\Delta y_{ex} = \alpha_{2,1} \times \Delta x_{m\prime} + \alpha_{2,2} \times \Delta y_{m\prime}$$

If we have multiple instances in of changes in position (like we do in between entries in the P_{ex} and P_m matrices) then we can rewrite this as a matrix form

$$\begin{bmatrix} \Delta x_{ex,1} & \Delta y_{ex,1} \\ \Delta x_{ex,2} & \Delta y_{ex,2} \\ \vdots & \vdots \\ \Delta x_{ex,n} & \Delta y_{ex,n} \end{bmatrix} = \begin{bmatrix} \Delta x_{m,1} & \Delta y_{m,1} \\ \Delta x_{m,2} & \Delta y_{m,2} \\ \vdots & \vdots \\ \Delta x_{m,n} & \Delta y_{m,n} \end{bmatrix} \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} \\ \alpha_{1,2} & \alpha_{2,2} \end{bmatrix}$$

This series of equations can be solved via least squares for the α terms as long as you have two non singular changes in position (aka three distinct positions in P_{ex} and P_m tha don't lie on a single line)

$p_{ex}(k)$ **parameters** We simply choose the last calibration position in P_{ex} to be the intial/reference position.

$$p_{ex}(k) = \begin{bmatrix} x_{ex,n} \\ y_{ex,n} \\ z_{ex,n} \end{bmatrix}$$

Putting it all together

We have a calibration model that defines where the location of the micropipette tip is in the external coordinate system when we know the tip's position in the manipulator coordinate system and we know where the tip started (the initial/reference position) external coordinate system and manipulator coordinate system. This calibration model states that the "new" tip position is the external coordinate system is the "old" tip positin in the external coordinate system plus its change in position (as transformed to the external coordinate system from the manipulator coordinate system).

$$p_{ex}(k+1) = p_{ex}(k) + \hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4} \times \Delta p_m(k)$$

We know the "old" position in the external coordinate system ($p_{ex}(k)$ is known), the "old" position in the manipulator coordinate system, and the "new" position in the manipulator system (so $\Delta p_m(k)$ is known)

We made some assumptions to define $\hat{T}_{m_3 \leftarrow m_4}$ in terms of the pipette angle, θ, which can be queried from the manipulator (so θ is known and therefore $\hat{T}_{m_3 \leftarrow m_4}$ is known).

$$\hat{T}_{m_3 \leftarrow m_4} = \begin{bmatrix} 1 & 0 & 0 & \cos(\theta) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \sin(\theta) \end{bmatrix}$$

We made some assumptions, to define the $\hat{T}_{ex \leftarrow m_3}$ matrix interms of 5 parameters. We assumed we knew the z-scaling, and we can use least squares regrssion on a list of non-singular external and manipulator positions to estimate the rest.

$$\hat{T}_{ex \leftarrow m_3} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 \\ \alpha_{2,1} & \alpha_{2,2} & 0 \\ 0 & 0 & \pm\alpha_z \end{bmatrix}$$

Altogether we have the following equation by expanding the terms and multiplying the matrices:

$$\begin{bmatrix} x_{ex}(k+1) \\ y_{ex}(k+1) \\ z_{ex}(k+1) \end{bmatrix} = \begin{bmatrix} x_{ex}(k) \\ y_{ex}(k) \\ z_{ex}(k) \end{bmatrix} + \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 & \alpha_{1,1} \times cos(\theta) \\ \alpha_{2,1} & \alpha_{2,2} & 0 & \alpha_{2,1} \times cos(\theta) \\ 0 & 0 & \pm\alpha_z & \pm\alpha_z \times sin(\theta) \end{bmatrix} \left(\begin{bmatrix} x_m(k+1) \\ y_m(k+1) \\ z_m(k+1) \\ d_m(k+1) \end{bmatrix} - \begin{bmatrix} x_m(k) \\ y_m(k) \\ z_m(k) \\ d_m(k) \end{bmatrix} \right)$$

Going the inverse (external to manipulator position)

What if we want the manipulator position given a the external position? For instance, we click on a cell we want to target and get its external coordinates, but how do we move the manipulator to that position?

We can rearange the equation above to get a useful relationship:

$$\Delta p_{ex}(k) = \hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4} \times \Delta p_m(k)$$

However, you can see above that $\hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4}$ is not square so it certainly can't be inverted to solve for $\Delta p_m(k)$. But we can do a little trick and say that we are going to hold one of the manipulator axes constant, so it can't move. Say that we aren't going to allow the injection axis to move so that $\Delta d_m = 0$. Then we can eliminate the "d" column in the $\hat{T}_{ex \leftarrow m_3} \times \hat{T}_{m_3 \leftarrow m_4}$ matrix, so that our our new equation looks like:

$$\Delta p_{ex}(k) = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & 0 \\ \alpha_{2,1} & \alpha_{2,2} & 0 \\ 0 & 0 & \pm\alpha_z \end{bmatrix} \times \left(\begin{bmatrix} x_m(k+1) \\ y_m(k+1) \\ z_m(k+1) \end{bmatrix} - \begin{bmatrix} x_m(k) \\ y_m(k) \\ z_m(k) \end{bmatrix} \right)$$

Now we can invert the matrix (assuming the modified version isn't singular) to solve for our change in manipulator position (for $\Delta d_m = 0$).