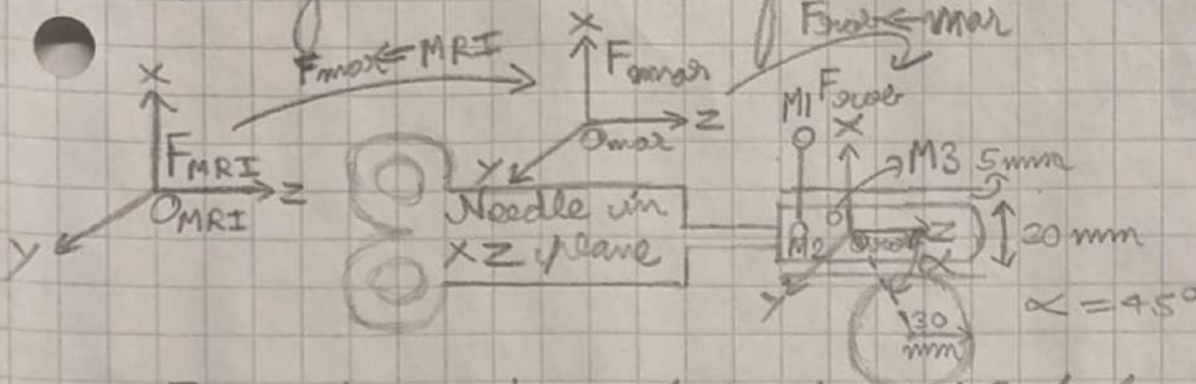
Navigation Transformation



→ Transformation of biopsy point from F_{MRI} to F_{rob} through F_{mar}

⇒ Frame transformation occurring $= F_{mar \Leftarrow MRI}$ & $F_{rob \Leftarrow mar}$

⇒ Frame transformation steps ⇒ ① Translation from O_{MRI} to O_{mar} by translation matrix $T_{mar \Leftarrow MRI}$ which contains vector $-O_{mar}$.

① $T_{mar \Leftarrow MRI} = \begin{bmatrix} 1 & 0 & 0 & -O_{marx} \\ 0 & 1 & 0 & -O_{mary} \\ 0 & 0 & 1 & -O_{marz} \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times4}$

② $R_{mar \Leftarrow MRI} = \begin{bmatrix} mar_{1x} & mar_{1y} & mar_{1z} & 0 \\ mar_{2x} & mar_{2y} & mar_{2z} & 0 \\ mar_{3x} & mar_{3y} & mar_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times4}$ ② Rotation of frame using $R_{mar \Leftarrow MRI}$

③ Since $F_{mar \Leftarrow MRI} = R_{mar \Leftarrow MRI} \cdot T_{mar \Leftarrow MRI}$,

$$F_{mar \Leftarrow MRI} = \begin{bmatrix} mar_{1x} & mar_{1y} & mar_{1z} & (mar_{1x} \cdot -O_{marx} + mar_{1y} \cdot -O_{mary} + mar_{1z} \cdot -O_{marz}) \\ mar_{2x} & mar_{2y} & mar_{2z} & (mar_{2x} \cdot -O_{marx} + mar_{2y} \cdot -O_{mar2y} + mar_{2z} \cdot -O_{marz}) \\ mar_{3x} & mar_{3y} & mar_{3z} & (mar_{3x} \cdot -O_{marx} + mar_{3y} \cdot -O_{mar3y} + mar_{3z} \cdot -O_{marz}) \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times4}$$

④ For $pt_{rob} = F_{mar \Leftarrow MRI} F_{rob \Leftarrow mar}\, pt_{MRI}$, we repeat steps ① → ③ for $F_{rob \Leftarrow mar}$ and get the overall frame transformation to plug into the equation here:

$F_{rob \Leftarrow MRI} = F_{mar \Leftarrow MRI} \cdot F_{rob \Leftarrow mar}$; so ↓

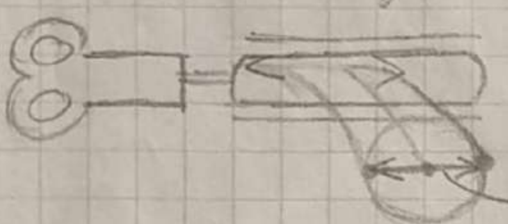
⑤ $pt_{rob} = F_{rob \Leftarrow MRI}\, pt_{MRI}$ // which is the transformation for the biopsy target point from F_{MRI} to F_{rob} through F_{mar}.

Workspace

Decoupled degrees of freedom

① Translate ② Rotate ③ Insert/Retract needle

⇒ Minimum range of motion for translation of end effector is 0 to 60 mm since the diameter of the prostate (assumed to be a sphere) is 60 mm, this can be seen from the diagram below:

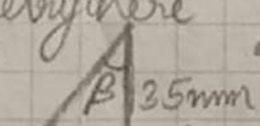


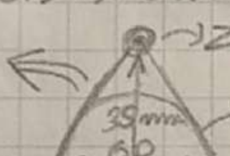
Effect range of translation = 60 mm

⇒ Minimum range of motion for rotation of end effector is calculated in the following way:

① Rotation occurs along the z axis; minimum range of motion should span 2 extreme lateral points furthest from the z axis shown below:

② We can do some trigonometry here to get ⇒





→ z-axis coming out of page

→ cross-section of sphere in the y-z plane

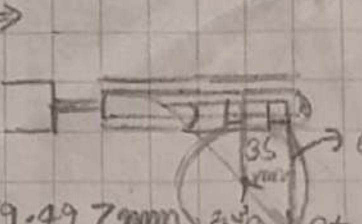
⇒ we know the hypotenuse is $\sqrt{35^2+30^2} \approx 46.097$

③ To find β, we do $\arcsin(30/46.097) = 40.601°$

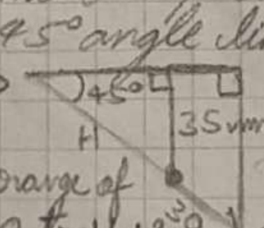
④ Thus we know that the min range of motion for rotation is $2\beta = \underline{81.202°}$

⇒ Minimum range of motion for insertion/rotation of needle is calculated as follows:

① The point furthest into the prostate would be straight down the diameter at a 45° angle like this ⇒



② We take this triangle ⇒ and H = 49.497°



③ Thus we know that min range of motion for needle is from 0 to H + 30 mm or 79.497 mm

Calibration

Needle deflection detection and quantification

→ Parts of this method are adapted from a study from 2020 titled "Needle deflection and tissue sampling in needle biopsy" conducted by Shih, A. et al. (cited in digital form)

Experimental Setup

→ Bring robot to its home position and orientation inside an MRI scanner (as it would be in a biopsy)

→ Reset all Degrees of freedom to nominal home positions; attach M4 marker to needle tip for tracking

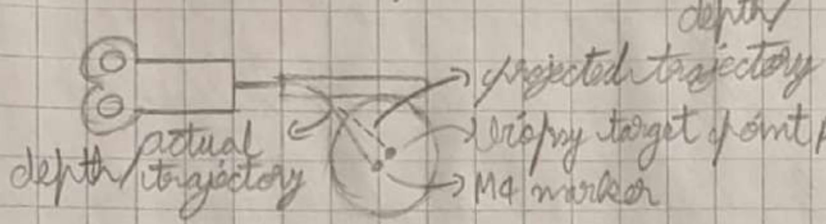
→ Position a multilayered 'phantom' to act as a prostate for experimentation; set biopsy target to be at the center of the 'prostate' in MRI frame

→ Set the robot to be rotated and translated halfway through their degrees of freedom and initialize needle insertion procedure with all markers being tracked.

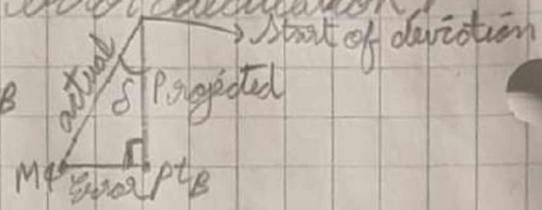
→ Execute needle insertion to hit predetermined biopsy target point and measure the position of the M4 marker at the end of insertion; also measure lateral distance from M4 to the biopsy point.

Li, A. D. R., Plott, J., Chen, L., Montgomery, J. S., & Shih, A. (2020). Needle deflection and tissue sampling length in needle biopsy. Journal of the mechanical behavior of biomedical materials, 104, 103632. https://doi.org/10.1016/j.jmbbm.2020.103632

Experimental setup sketch



Resulting triangle for error calculation:



→ Here, measuring the difference between coordinates for M4 and pt_B gives us the error distance which we can use to sub into trig rules for right angled triangles ⇒ $\frac{\sin \delta}{\text{Error}} = \frac{\sin 90^\circ}{\text{Actual depth}}$ ⇒ which gives us the angle of deflection at that depth

→ We repeat the whole process to find the 'Error' and 'δ' for various insertion depths ⇒ a good way to do this may be to take the 6 extreme points on the prostate in all the different orthogonal axes (rotated 45° to suit projected trajectory) and plot the needle deflection angle and error distance between M4 and pt_B at different insertion depths

→ If the plots described above are acceptable under minimum threshold conditions, we would accept the needle for actual use, otherwise reject and send it for heat treatment.

Calibration

Robot frame F_{rob} calibration with respect to F_{mri}

→ We initialize the setup to calibrate the robot frame by placing the robot in the nominal home position inside the MRI scanner frame as we would in a real biopsy.

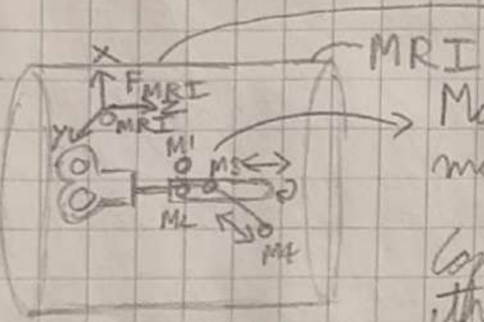
→ We use markers M1, M2, M3 and M4 as we move the robot along its degrees of freedom in our given ranges: ① Translation along the z axis in F_{MRI} frame observing M1, M2 and M3
② Rotation about the z axis cycling 360° ideally observing M1, M2 and M3
③ Extending and retracting needle with M4 through its allowed range of motion to find α, capturing all motion in F_{MRI}.

→ Observed motion in F_{MRI} frame helps compute F_{mri} using normalized bases which are transformed to F_{rob} in conjunction with the α made in the xz plane.

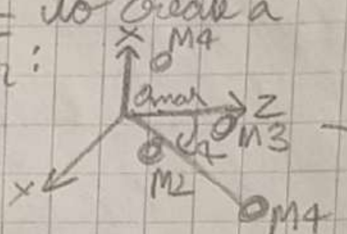
→ Observed coordinates of M1, M2 and M3 are transformed to F_{rob} using rigid body transformation and alpha is kept constant throughout. (we use an SVD algorithm for rigid body transformation)

→ This process is repeated numerous times to minimize error

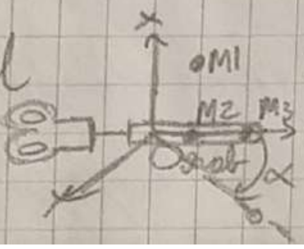
Robot frame calibration setup



Markers into F_{MRI} to create a marker frame F_{mar}:



Converted then to the F_{rob} frame after returning robot to nominal home position



Marker coordinates transformed to F_{rob}

Kinematics

Inverse Kinematics

Computing values of translation, rotation and needle insertion that will take the needle tip to the desired point from its home position.

→ A desired biopsy target point pt_B in frame F_{MRI} would be transformed to F_{rob} through F_{mar} where translational and rotational differences are resolved

⇒ Translation along z-axis would simply be the z axis coordinate

⇒ Rotation about the z-axis would be arctan(y-coord, x-coord)

⇒ Insertion depth can be calculated by finding the distance from the origin of the robot frame to the the target point in the XY plane.

① effective_reach = $\sqrt{x_{pt}^2 + y_{pt}^2}$

To account for the angle of insertion we find true depth

② insertion = effective_reach / cos(α); $\alpha = 45°$

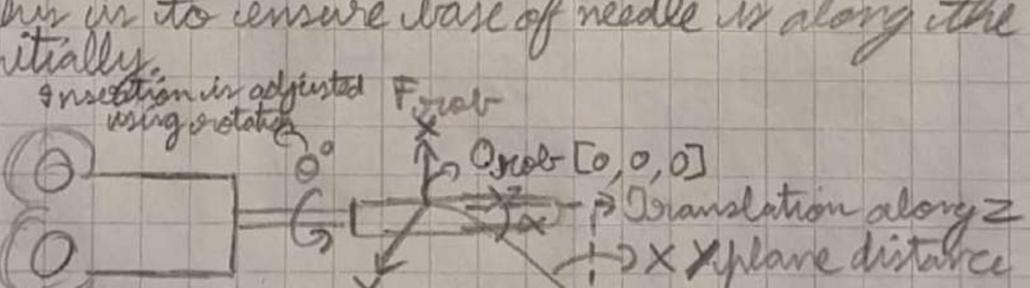
*

⇒ Translation needs to be adjusted to account for vertical movement which is first calculated:

① vertical_comp = insertion * sin(α)

So ② Translation = Translation − vertical_comp

⇒ This is to ensure base of needle is along the z axis initially



Kinematics

Forward Kinematics

Computing the resulting location of needle tip upon moving the motion stages (translation, rotation, insertion) from its home position

→ The desired position of the needle tip is calculated on the combined effect of these motions:

⇒ Translation is on z axis so translation would create a $(0, 0, T_z)$ shift
⇒ Rotation affects X and Y coords of needle tip in F_{ro-}
⇒ Insertion is the factor the above are tied together ⇓
① We decompose the needle insertion into XY plane and Z-axis components

→ insert_xy = insertion × cos(α) # Projection onto XZ plane

→ insert_z = insertion × sin(α) # Vertical component

→ Needle tip in F_{rob} frame
⇒ x_comp = insert_xy × cos(rotation)
⇒ y_comp = insert_xy × sin(rotation)
⇒ z_comp = translation × insert_z