🤖

Kinematics in Medical Robotics

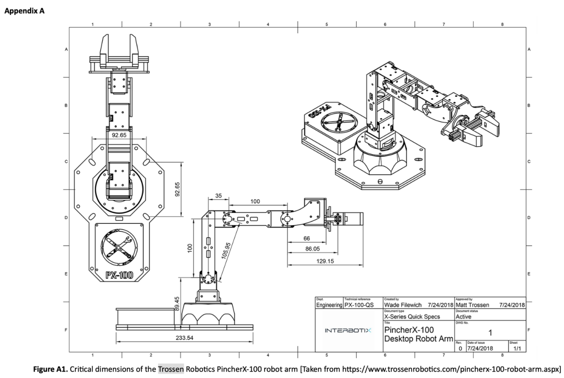
▼ **Table of Contents**

Introduction 👋

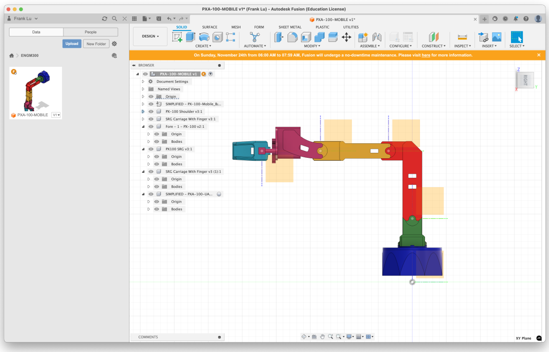
As medical robots continue to offer unprecedented precision and consistency in challenging procedures, understanding the kinematic foundations of these systems becomes crucial for optimizing their medical applications. This report starts off with a detailed kinematic analysis of the much simpler 4DOF Trossen PincherX 100 robotic arm, examining its joint parameters and link variables. Through the derivation of the Denavit-Hartenberg parameters and transformation matrices for each link, I establish a simulation framework for analyzing the robot's workspace and kinematic limitations. In the end, I contrast these findings with the

CyberKnife®, a much more sophisticated 6DOF medical robot specifically designed for radiation therapy. By examining how kinematic configurations for different robots affect performance in the clinical setting, I demonstrate the critical relationship between degrees of freedom, workspace characteristics and medical efficacy.

Throughout this report, I will be using the custom simulations and robot models I've built. If you'd like to experiment with the interactive demos, I recommend cloning my code repository and downloading the modified model shared on Fusion Team [3].



Engineering drawing by Trossen Robotics [1]

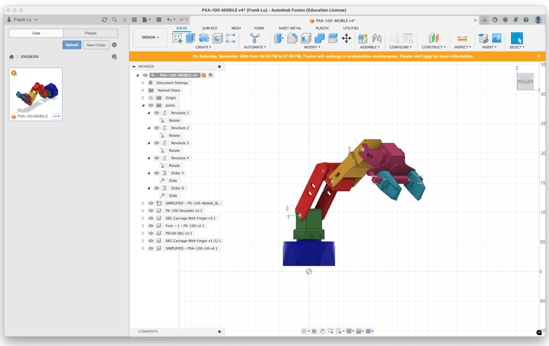


PincherX-100 Solid STEP Files by Interbotix [1]

For reference, all simulations and model files are built using the engineering drawing shown on the left. Moreover, Interbotix has provided detailed specifications—including joint limits and the original model STEP files—here. I have incorporated the joints and their joint limits from the table below into the Interbotix model using Autodesk Fusion. As noted above, you can explore the arm's full range of motion with my publicly available model on Fusion Team here [3].

Joint	Min	Max	Servo ID(s)
Waist	-180	180	1
Shoulder	-111	107	2
Elbow	-121	92	3
Wrist Angle	-100	123	4
Gripper	30mm	74mm	5

Default joint limits provided by Interbotix [1]

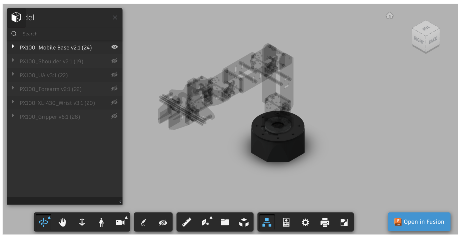


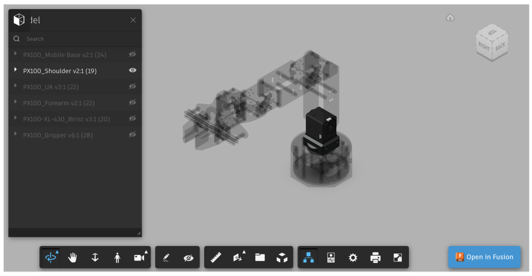
The PincherX 100 model with fully functional joints and joint limits [3]

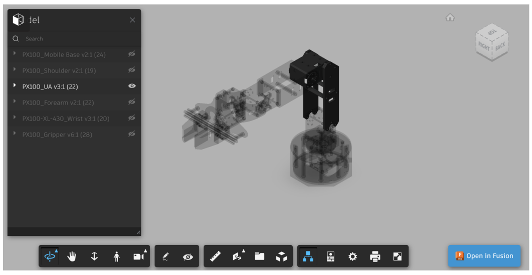
PincherX 100 🦾

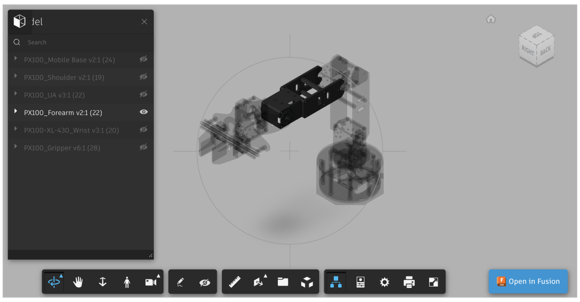
The Trossen PincherX 100 consists of 5 links, as shown in the Fusion model provided by the Interbotix Arms Documentation [1]. The robot's structure is composed of the following joints and links:

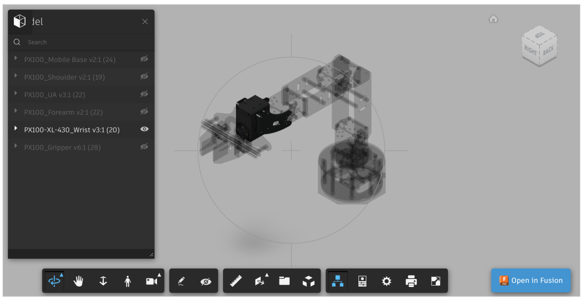
- **Waist**: The base of the robot (link 0), featuring a revolute joint that rotates around the z-axis.
- **Shoulder**: The second link (link 1), connected to the waist via a revolute joint.
- **Elbow**: The third link (link 2), connected to the shoulder via a revolute joint.
- **Forearm**: The fourth link (link 3), connected to the elbow via a revolute joint.
- **Wrist**: The final link (link 4) that attaches to the end effector.

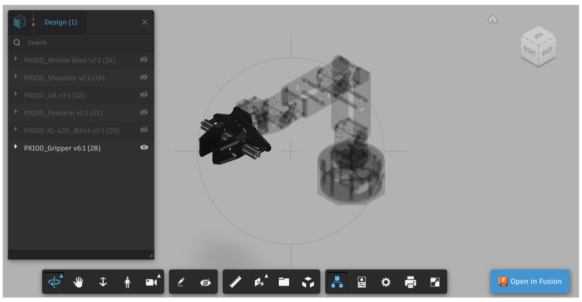

Waist (base + revolute joint 1) [1]


Shoulder (link 1 + revolute joint 2) [1]


Elbow (link 2 + revolute joint 3) [1]


Forearm (link 3 + revolute joint 4) [1]


Wrist (link 4) [1]


Gripper (end effector) [1]

Using Grübler's formula for mobility, we can determine that the Trossen Pincher X Robot has **4 degrees of freedom**, as verified on the Interbotix X-Series Arms Documentation.

$$M = 6(L - N - 1) + \sum_j f_j$$

$$M = 6(5 - 4 - 1) + 4(1)$$

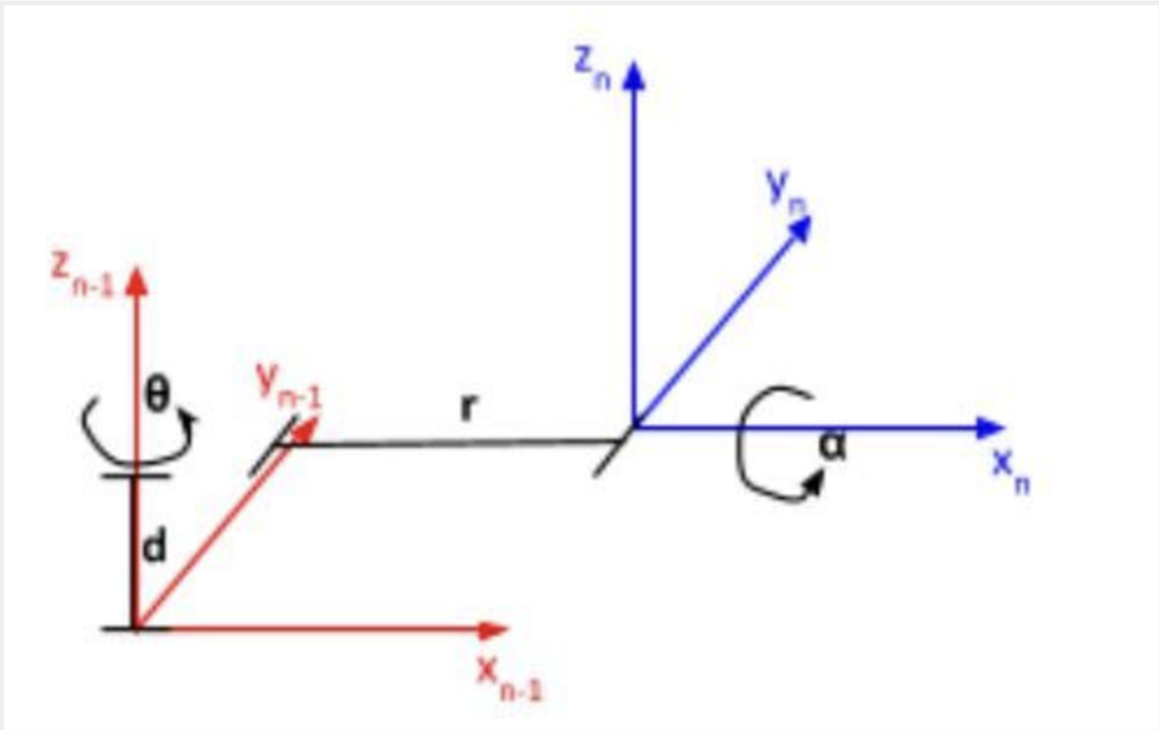
$$M = 4$$

- M = Mobility (degrees of freedom)
- L = Number of links (including base/ground)
- N = Number of joints
- f_j = Degrees of freedom for each joint

The formula multiplies by 6 to account for the degrees of freedom of a rigid body in space.

Link Parameters and Joint Variables

Identifying the link parameters and joint variables is relatively straightforward after figuring out the links and joints. Recall that the Denavit-Hartenberg (DH) parameters consist of four values for each link:



DH Parameters Visualization by Addison [2]

$$\theta = \angle(x_{n-1}, x_n) \text{ around } z_{n-1}$$

$$\alpha = \angle(z_{n-1}, z_n) \text{ around } x_n$$

$$a = \text{distance}(O_{n-1}, O_n) \text{ along } x_n$$

$$d = \text{distance}(x_{n-1}, x_n) \text{ along } z_{n-1}$$

Rotational Variables

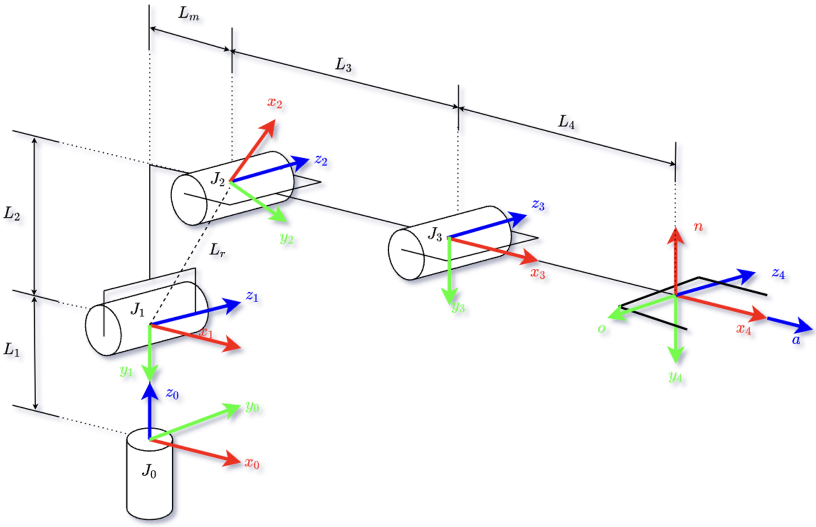
- θ: The joint angle about the z-axis.
- α: The angle about the common normal, from the old z-axis to the new z-axis.

Displacement Variables

- a: The length of the common normal (or radius from previous z-axis to new z-axis)
- d: The offset along the previous z-axis to the common normal

Denavit-Hartenberg Parameters

Using the provided engineering drawing, we can systematically go through each link, from the base to the end-effector, to determine the link parameters and joint variables and construct the following DH table.



Robot description provided by Chitiva [5]

Link i	α_{i-1} (deg)	a_{i-1} (mm)	θ_i (degrees)	d_i (mm)
1 (Base to Waist)	0	0	θ_1 (-180° to 180°)	L1 = 89.45
2 (Waist to Shoulder)	-90	Lr = 105.95	θ_2 (-111° to 107°)	0
3 (Shulder to Elbow)	0	L3 = 100	θ_3 (-121° to 92°)	0
4 (Elbow to Wrist)	0	L4 = 66	θ_4 (-100° to 123°)	0

💡 **Modified (Proximal) DH Parameters**

The DH parameters in the table follow a specific sequence: first transformations around the x-axis (displacement and rotation), then transformations around the z-axis. This matches the convention taught in the lecture material and detailed in *Introduction to Robotics: Mechanics and Control*. See the section below for additional context.

Note that for the elbow, we calculate the diagonal length using the Pythagorean theorem:

$$L_r = \sqrt{L_m^2 + L_2^2}$$

because the diagonal represents the direct distance of our z-axis shift. The rotation around the z-axis from x_1 to x_2 is already accounted for in our rotational parameter θ_2.

Additionally, since joints J_0 and J_1 are aligned, we **could simplify the DH parameters by setting our base frame at the first joint**. When two consecutive joint axes (z-axes) are aligned, their rotation axes coincide. This means that rotating around either axis will produce the same type of motion. **However, the current approach provides more explicit documentation of the robot's structure.**

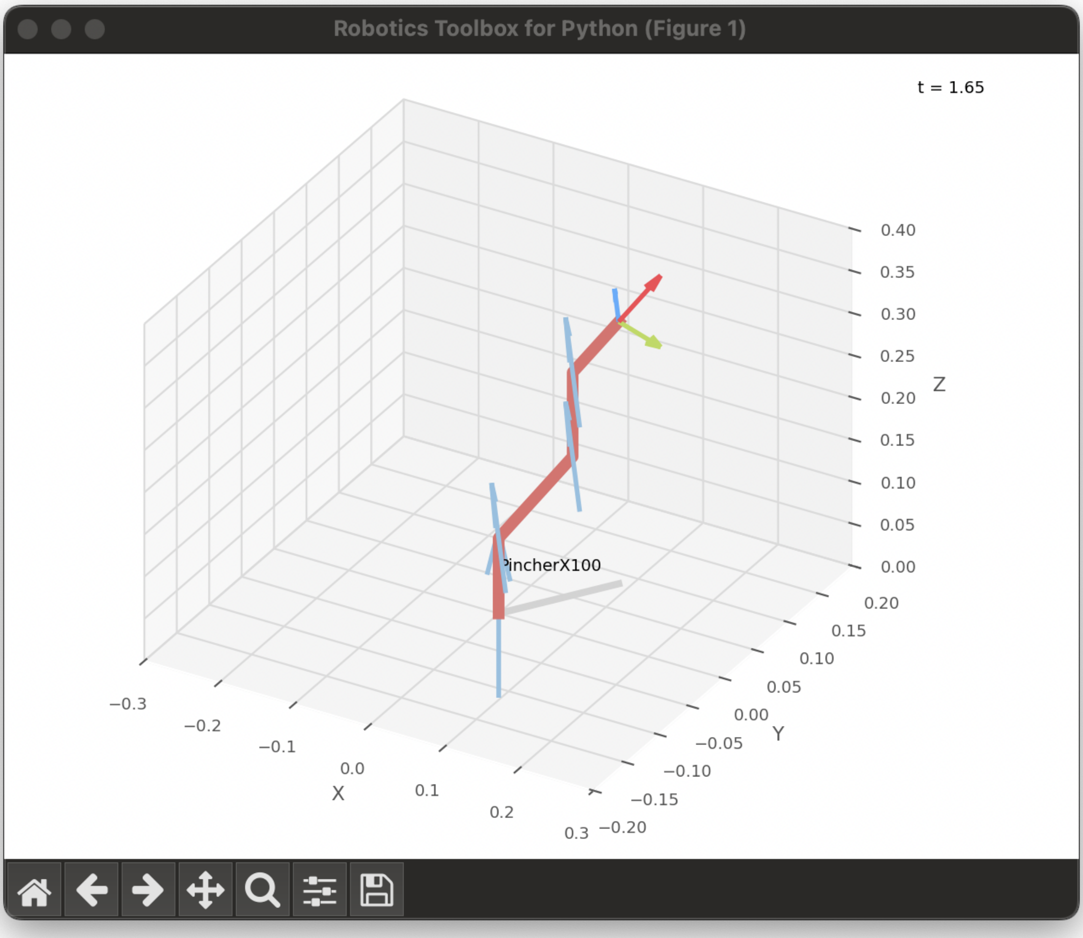
Kinematics Simulation

Using Peter Corke's robotics-toolbox-python, I have created a robot model to verify the DH parameters above. Keep in mind that the DH parameter output below displays each link's variables relative to its own frame, rather than the previous link's frame. You can find the simulation code here.

```
DHRobot: PincherX100, 4 joints (RRRR), dynamics, standard DH parameters
```

θ_j	d_j	a_j	α_j	q^-	q^+
q1	0.08945	0	-90.0°	-180.0°	180.0°
q2	0	0.106	0.0°	-111.0°	107.0°
q3	0	0.1	0.0°	-121.0°	92.0°
q4	0	0.066	0.0°	-100.0°	123.0°

Output of the DH parameters upon initialization [3]



Simulation of a pre-defined trajectory of the robot [3]

Transformation Matrices for Each Link

The transformation matrix can be expressed as the product of four basic transformations: twist, link length, rotation, and offset, as shown in week 3's lecture [3]:

$$ {}_{i}^{i-1}T = R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)D_z(d_i) $$

Where:

$$ R_x(\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ D_x(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$R_z(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D_z(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The multiplication gives us the final matrix

$${}^{i-1}_{i}T = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\theta_i)\cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\theta_i)\sin(\alpha_{i-1}) & \cos(\theta_i)\sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the above matrix, we can calculate

Transformation Matrix from Base to Waist

$${}^{0}_{1}T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1)\cos(-90^\circ) & \cos(\theta_1)\cos(-90^\circ) & -\sin(-90^\circ) & -89.45\sin(-90^\circ) \\ \sin(\theta_1)\sin(-90^\circ) & \cos(\theta_1)\sin(-90^\circ) & \cos(-90^\circ) & 89.45\cos(-90^\circ) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which gives us

$${}^{0}_{1}T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 89.45 \\ -\sin(\theta_1) & -\cos(\theta_1) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation Matrix from Waist to Shoulder

$${}^{1}_{2}T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 105.95 \\ \sin(\theta_2)\cos(0) & \cos(\theta_2)\cos(0) & -\sin(0) & -0\sin(0) \\ \sin(\theta_2)\sin(0) & \cos(\theta_2)\sin(0) & \cos(0) & 0\cos(0) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

which gives us

$${}^{1}_{2}T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 105.95 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformation Matrix from Shoulder to Elbow

$$
{}^{2}_{3}T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 100 \\ \sin(\theta_3)\cos(0) & \cos(\theta_3)\cos(0) & -\sin(0) & -0\sin(0) \\ \sin(\theta_3)\sin(0) & \cos(\theta_3)\sin(0) & \cos(0) & 0\cos(0) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

which gives us

$$
{}^{2}_{3}T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 100 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Transformation Matrix from Elbow to Wrist

$$
{}^{3}_{4}T = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 66 \\ \sin(\theta_4)\cos(0) & \cos(\theta_4)\cos(0) & -\sin(0) & -0\sin(0) \\ \sin(\theta_4)\sin(0) & \cos(\theta_4)\sin(0) & \cos(0) & 0\cos(0) \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

which gives us

$$
{}^{3}_{4}T = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 66 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Combining the Transformations

To find the overall transformation from the base to the end-effector, we multiply the matrices sequentially:

$$
T =^{0}_{1} T \cdot^{1}_{2} T \cdot^{2}_{3} T \cdot^{3}_{4} T
$$

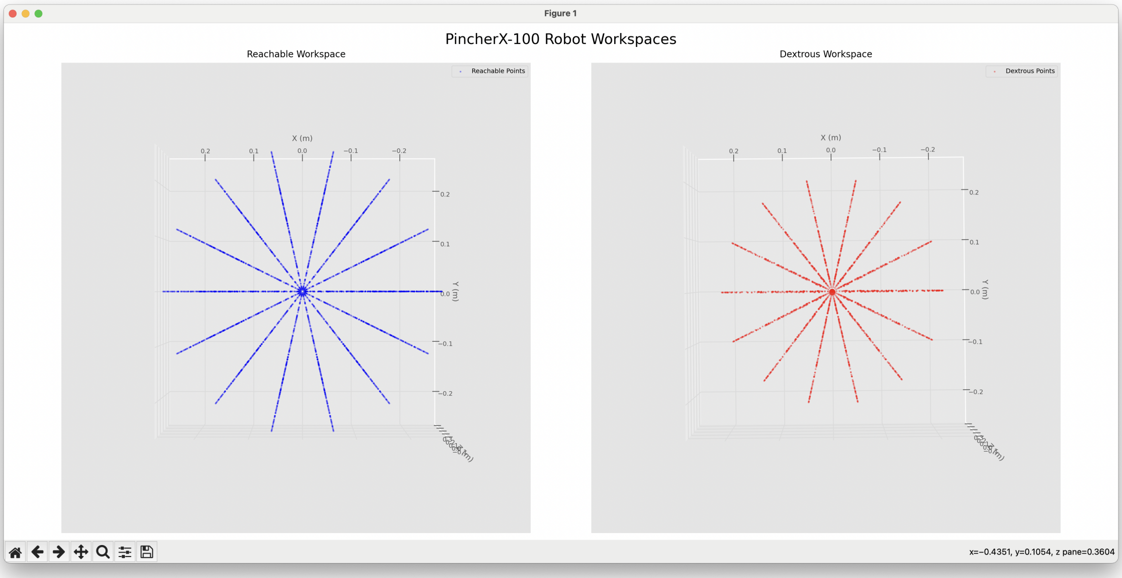
Workspace Exploration

Given that we are working with a 4-DOF robot, the configuration space is 4-dimensional. Each dimension corresponds to one degree of freedom made up of the joint angles for the 4 revolute joints. Using the simulation I've built here, we will explore this configuration's 3D workspace **subspace** where the end-effector operates.

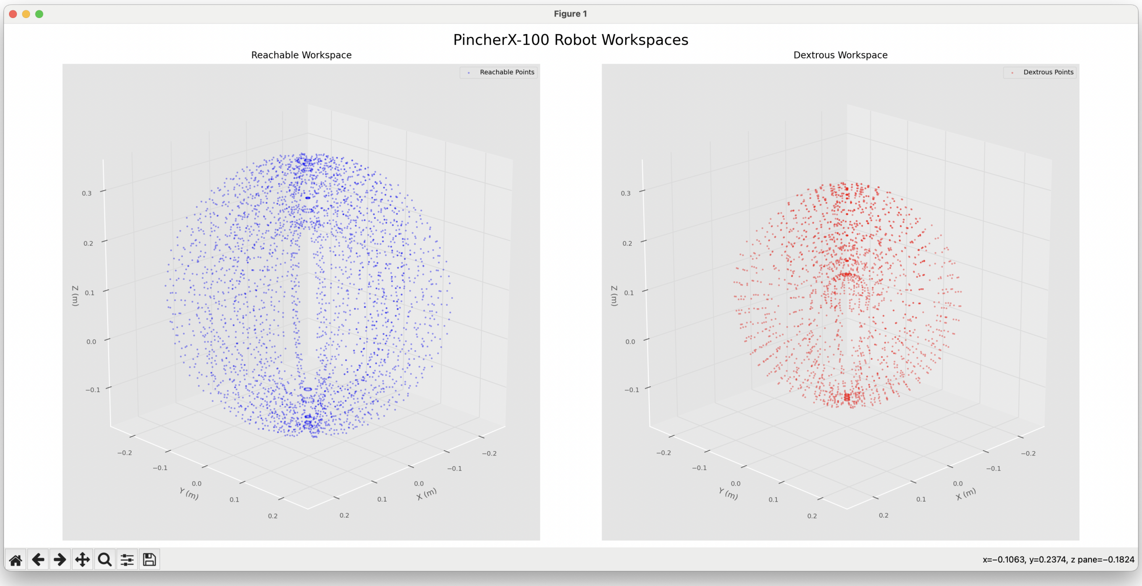
$$
f(\theta_1, \theta_2, \theta_3, \theta_4) \rightarrow (x, y, z) \in \mathbb{R}^3
$$

The first three joints determine the end effector's position (x, y, z), while the fourth joint only controls the wrist's pitch angle. Although the end effector includes a gripper that acts like a prismatic joint, **it is not considered a true kinematic degree of freedom since it affects neither position nor orientation**. The gripper's opening range simply determines what objects it can grasp, without affecting the robot's workspace dimensions.

Using the transformation matrix above, we can plot the robot's workspace in 3D space. The plot below shows both the reachable workspace (left) and dextrous workspace (right). You can find the code in the `plot_workspace` function here.



Top view of the workspaces [3]



Home view of the workspaces [3]

Reachable Workspace

The reachable workspace is the total volume of space that the robot's end-effector can reach **in at least one orientation**. This includes all points that the end-effector can

Dextrous Workspace

The dextrous workspace is where the end-effector can reach points **with any orientation within its rotational limits**. This space represents where the robot has full reach regardless of its orientation. The robot's reachable workspace forms a spherical shell due to its link lengths and joint constraints. While the base can rotate in a complete circle horizontally, joints 2, 3, and 4 limit the vertical reach. At the workspace boundary, the arm reaches its maximum extension from the base.

manipulability and can approach targets from multiple angles. As expected, **the dextrous workspace appears smaller because it is a subset of the reachable workspace**. The density of points is higher **in regions where the robot has better manipulability**.

Every point in the dextrous workspace must also be in the reachable workspace, but not vice versa. This means that while the robot might be able to reach a point, it may not be able to perform tasks requiring orientation changes at that point. Note that **the small semisphere appears more prominent in the dextrous workspace** visualization due to our calculation method (See code below). The reachable workspace plot uses a single point per joint angle combination with a fixed wrist angle ($\theta 4 = 0$), resulting in a more uniform distribution of points throughout the space.

💡 **Simulation Limitation!**

Our simulation models the robot as a simple kinematic chain with zero-thickness links—essentially a stick figure drawing. However, the actual PincherX-100 has:

- Physical link thicknesses that could cause collisions
- Motors and servos at each joint that take up space
- Cable routing and other mechanical components
- Actual joint mechanisms that might restrict movement in ways not captured by simple angle limits

Additionally, the **star pattern** ⭐ visible in the top views is a results of our sampling method. By dividing joint 1's range of [-180, 180] (360°) into 14 equal sections with 15 samples (including endpoints), we create "rays" that are spaced approximately 25.7° apart. If we wanted a smoother, less star-like pattern, we could **increase our samples.**

Workspace Code Explained

```
# Calculate reachable workspace points
reachable_points = []
for t1 in theta1:
    for t2 in theta2:
        for t3 in theta3:
            T = robot.fkine([t1, t2, t3, 0])
            if T is not None:
                pos = [float(T.t[0]), float(T.t[1]), float(T.t[2])]
                reachable_points.append(pos)
```

```
reachable_points = np.array(reachable_points)
```

Code for reachable workspace [3]

This code calculates all positions the robot's end-effector can reach in at least one orientation. For each combination of base (t1), shoulder (t2), and elbow (t3) angles, we only need to demonstrate one valid wrist angle to prove a point is reachable. That's why we use the **neutral wrist angle of 0**. Each point in this workspace indicates a position the robot can reach, though it may only be able to do so in one specific orientation.

```
# Calculate dextrous workspace points
dextrous_points = []
for t1 in theta1:
    for t2 in theta2:
        for t3 in theta3:
            is_dextrous, reference_position = is_point_dextrous(
                robot, t1, t2, t3, theta4
            )
            if is_dextrous:
                dextrous_points.append(reference_position)
dextrous_points = np.array(dextrous_points)
return reachable_points, dextrous_points
```

Code for dextrous workspace [3]

```
def is_point_dextrous(robot, t1, t2, t3, theta4, position_threshold=0.0
5, angle_threshold=0.2):
    # 1. Check for elbow singularities (both fully extended and folded
back)
    arm_angle = abs(t2 + t3)
    if abs(arm_angle) < angle_threshold or abs(arm_angle - np.pi) < ang
le_threshold:
        return False, None

    # 2. Check for vertical alignment singularity
    if (abs(abs(t2) - np.pi/2) < angle_threshold) and (abs(t3) < angle_
threshold):
        return False, None

    # Collect positions for different wrist angles
    positions = []
    for t4 in theta4:
        T = robot.fkine([t1, t2, t3, t4])
        if T is not None:
            pos = [float(T.t[0]), float(T.t[1]), float(T.t[2])]
```

```
            positions.append(pos)

    # 3. Check wrist dexterity through position variation
    if len(positions) > 0:
        positions = np.array(positions)
        variation = np.max([np.std(positions[:, i]) for i in range(3)])
        if variation < position_threshold:
            return True, positions[0]

    return False, None
```

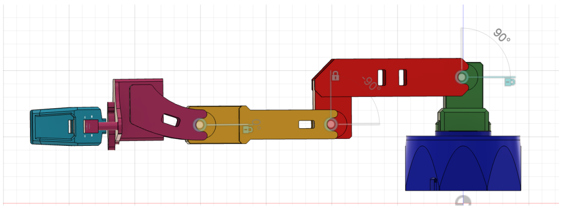
Code for checking dexterity of different wrist orientations [3]

This code here is a bit more demanding. For each combination of base, shoulder, and elbow angles, we must test every possible wrist angle since **a point qualifies for the dextrous workspace only if the robot can reach it at all possible wrist orientations**. I have written the `is_point_dextrous` function that does the following:

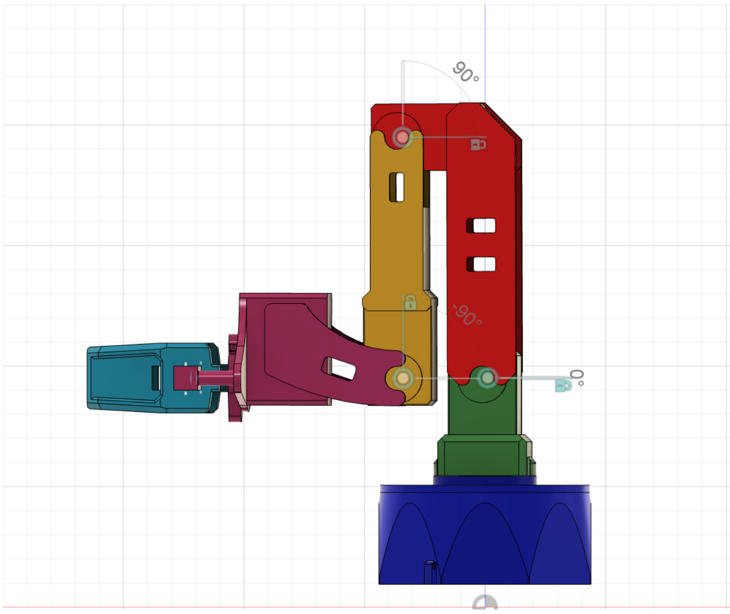
1. **Singularities Check**: The code first checks for singularity positions by examining elbow singularities (through joints 2 and 3) and vertical alignment. It discards the points if it is in a singularity position.
2. **Wrist Position Testing**: For each arm configuration, it tests all possible wrist angles while keeping other joint positions fixed. The code stores the end-effector positions calculated by the forward kinematics function `fkine` .
3. **Position Analysis**: Finally, it analyzes how much the end-effector position varies as the wrist moves. Using a 5 cm threshold, if the standard deviation of positions stays below this limit, we treat these positions as effectively identical. This threshold can be adjusted as needed.

Kinematic Limitations

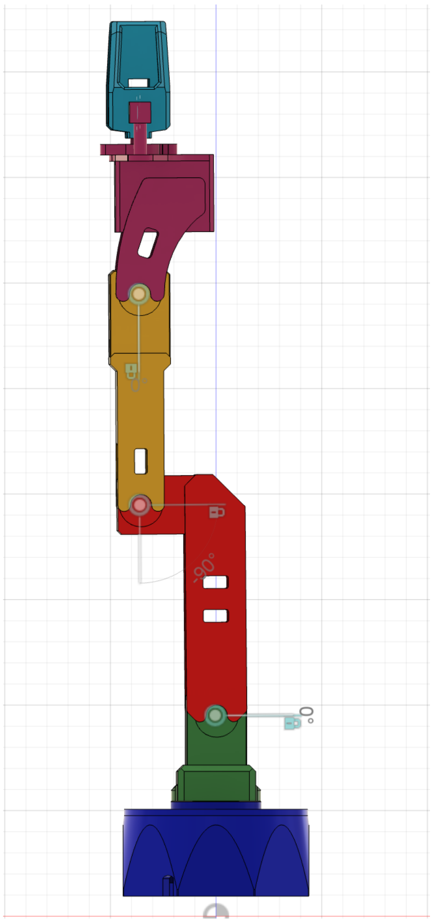
Having analyzed the workspace, I am going to explain how my code checks for singularities when calculating the reachable workspace. While I won't delve into the mathematical details of the Jacobian matrix for these configurations—as that's beyond this assignment's scope—I'll explain the intuitive understanding of these singularities below. Unlike 6-DOF manipulators which experience shoulder, elbow, and wrist singularities (explained more in the CyberKnife® Section), the PincherX-100's 4-DOF structure and simple **pitch-only wrist** means it only encounters modified versions of these. **Without a spherical wrist, it can't have wrist singularities**, its shoulder singularities appear as our **vertical alignment singularities**, and the elbow singularities manifest as our **fully extended and folded back configurations**.



Fully extended singularity [3]



Folded back singularity [3]



Vertical Alignment Singularity [3]

Fully Extended Singularity (Top Left) occurs when joints 2 and 3 align to create a straight line between their links, regardless of the overall arm orientation. In this configuration, the robot loses its ability to make small movements perpendicular to this line because both joints would need to move simultaneously to break out of this alignment. For surgical applications, this would severely limit the robot's ability to make delicate movements or **respond to small position corrections** when fully extended.

Folded Back Singularity (Bottom Left) happens when joints 2 and 3 align in the opposite direction, effectively folding the arm back on itself. This internal singularity creates a situation where the robot loses control in directions perpendicular to the plane of the fold. The configuration couples the motion of these joints, **similar to the fully extended case but in a folded configuration**. In practical terms, this would **prevent the robot from making fine adjustments when working in confined spaces** or when needing to reach behind obstacles.

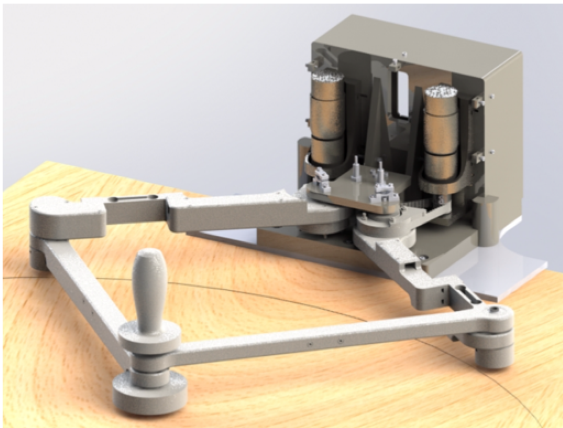
Vertical Alignment Singularity occurs when the robot arm points nearly straight up or down, causing the base rotation (joint 1) and wrist pitch (joint 4) to become redundant. In this configuration, rotating the base or pitching the wrist would create similar arcs of motion at the end-effector. This loss of independent control is particularly problematic for overhead operations or tasks requiring the robot to work directly below its mounting point. For instance, in a ceiling-mounted surgical setting, this singularity would limit the robot's ability to make precise movements **when working directly below its base**, potentially affecting critical vertical approaches to the surgical site

Finally, the robot faces one more limitation that isn't a true singularity but rather **a wrist design constraint**. The PincherX-100's wrist can only perform pitch movements (θ4), limiting end-effector orientation to a single rotation plane. This design severely restricts the robot's ability to achieve varied tool orientations. Though the manufacturer offers custom 3D-printed end-effectors, these can only partly overcome this fundamental limitation.

Medical Applications

The lack of full pose control limits the robot's ability to perform tasks requiring precise tool orientations—such as suturing at specific angles or approaching anatomical structures from optimal directions during laparotomy. Dead zones in the workspace may force awkward positioning of the robot relative to the patient. Additionally, the limited dextrous workspace restricts the robot's ability to maintain consistent tool control when navigating curved anatomical surfaces or operating through constrained access ports in minimally invasive surgery.

I mean, it's pretty much an excavator 🤷‍♂️ [8]



Neurorehabilitation planar robot [9]

Accounting for the analysis above, the Pincher X100 is best suited for non-critical support tasks in medical robotics. Despite its limitations, with task-specific modifications, **the robot can perform auxiliary tasks like cleaning, maintenance, and bedside assistance**. For example, the arm can deliver lightweight objects to patients and staff. Another potential application is rehabilitation through pre-planned trajectories, similar to the planar parallel device for neurorehabilitation discussed in lecture and shown above on the right.

The CyberKnife® Treatment Delivery System 🔪

The CyberKnife® System features a robot-mounted linear accelerator that delivers high-energy x-rays or photons for radiation therapy. The robot's ability to bend and move around the patient enables radiation delivery from thousands of distinct beam angles. This flexibility greatly expands the possible positions for concentrating radiation on tumors **while minimizing**

exposure to surrounding healthy tissue. Through its robotic delivery and real-time image guidance, the system has established new standards for precision in stereotactic radiosurgery (SRS) and stereotactic body radiation therapy (SBRT) across all tumor types.

The CyberKnife® [5]

KUKA KR Quantec [6]

The CyberKnife® robotic radiosurgery system **is built on a KUKA-based robotic manipulator with 6-DOF**. Unlike the Trossen robot's simpler 4-DOF serial chain configuration, the CyberKnife® employs a more complex kinematic structure. The 6-DOF system enables **complete spatial positioning and full orientation control**. This allows the end effector to perform **roll, pitch, and yaw** movements while maintaining precise positioning during operation.

Kinematic Simulation

Since our robotics toolbox library only includes the KR5 model, we will approximate the CyberKnife® (which is actually based on the KUKA KR Quantec) using the KR5 arc model. The CyberKnife® features a sophisticated joint arrangement enabling precise and flexible 3D movement. Its shoulder has a significant joint offset (d1), and its substantial second link length provides extended reach. Compared to the Trossen PincherX 100, the CyberKnife® consists of:

- A base revolute joint (θ_1) with a large vertical offset
- Three joints (θ_2, θ_3, θ_4) that form a spherical shoulder/elbow complex
- Two final joints (θ_5, θ_6) creating a spherical wrist

These features create a large spherical workspace—ideal for moving around a patient during radiation therapy. It can maintain a consistent focus point while moving around the treatment areas from multiple angles while avoiding obstacles. To summarize, while the PincherX 100 is more **planar-dominated**, the CyberKnife® is truly **omnidirectional**.

The DH parameter table from the imported KR5 robot is shown below.

```
DHRobot: KR5 (by KUKA), 6 joints (RRRRRR), dynamics, geometry, standard
DH parameters

┌─────┬────────┬─────┬───────┬────────┬───────┐
│ θⱼ  │   dⱼ   │ aⱼ  │  ⍺ⱼ   │   q⁻   │  q⁺   │
├─────┼────────┼─────┼───────┼────────┼───────┤
```

q1	0.4	0.18	-90.0°	-155.0°	155.0°
q2	0	0.6	0.0°	-180.0°	65.0°
q3	0	0.12	90.0°	-15.0°	158.0°
q4	-0.62	0	-90.0°	-350.0°	350.0°
q5	0	0	90.0°	-130.0°	130.0°
q6	-0.115	0	180.0°	-350.0°	350.0°

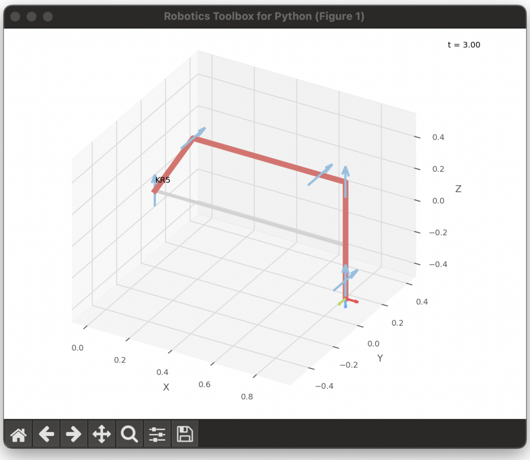
name	q0	q1	q2	q3	q4	q5
qr	45°	60°	45°	30°	45°	30°
qz	0°	0°	0°	0°	0°	0°
qk1	45°	60°	45°	30°	45°	30°
qk2	45°	60°	30°	60°	45°	30°
qk3	30°	60°	30°	60°	30°	60°

The DH parameters of the Kuka KR5 [3]

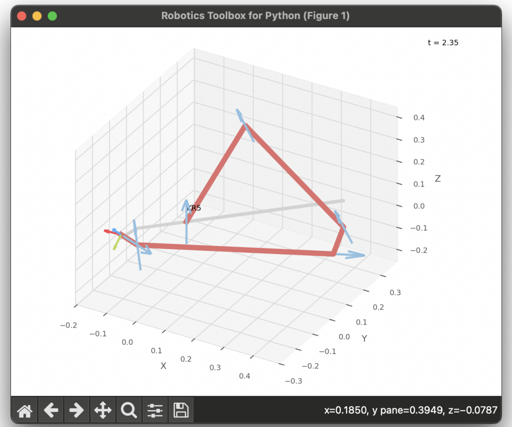
Conveniently, the toolbox includes several preset configurations—`qr`, `qz`, `qk1`, `qk2`, and `qk3`—that are ready for visualization. It's safe to deduce that the `qz` configuration is the zero angle configuration, probably used as a reference position with **all joint angles at 0 degrees used for calibration**. The `qr` configuration is the **standby or ready configuration** with specific joint angles that minimizes singularity risks while maximizing maneuverability. The `k` configurations demonstrate different robot poses and capabilities:

- `qk1` matches the ready position (qr), serving as a baseline for motion planning.
- `qk2` shows kinematic redundancy by reaching similar positions with different joint angles. Note the change in the elbow and wrist joints while maintaining the same end effector positions.
- `qk3` uses symmetrical joint angles (30° and 60°), which might be useful for demonstrating the robot's balanced movement capabilities.

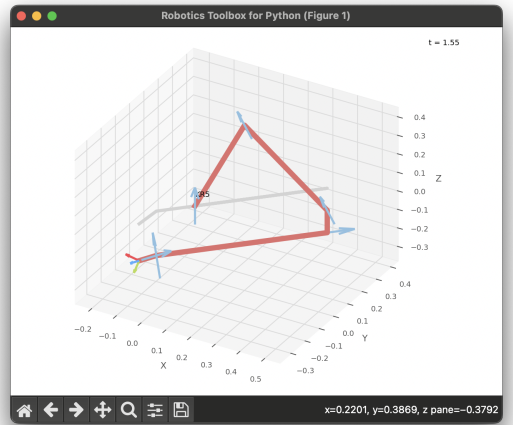
Below you can find the screenshots of my simulation in different configurations and you will be able to see the full trajectory code here.



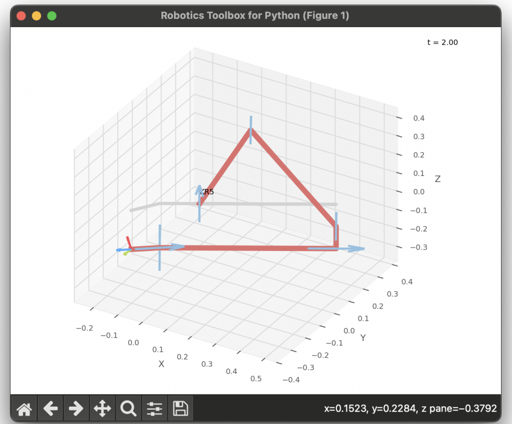
The `qz` configuration



The `qr` / `qk1` configuration



The `qk2` configuration



The `qk3` configuration

```
from roboticstoolbox.models.DH import KR5
import numpy as np

# Create the robot
robot = KR5()

# Show each configuration individually
configs = [robot.qz, robot.qr, robot.qk1, robot.qk2, robot.qk3]
config_names = ['qz', 'qr', 'qk1', 'qk2', 'qk3']

# Display each configuration
for config, name in zip(configs, config_names):
    print(f"\nDisplaying {name} configuration...")
    print(f"Joint angles: {np.rad2deg(config)} degrees")
    robot.plot(config, block=True)
```

```
# Now show the full trajectory
print("\nNow showing full trajectory through all configurations...")
```

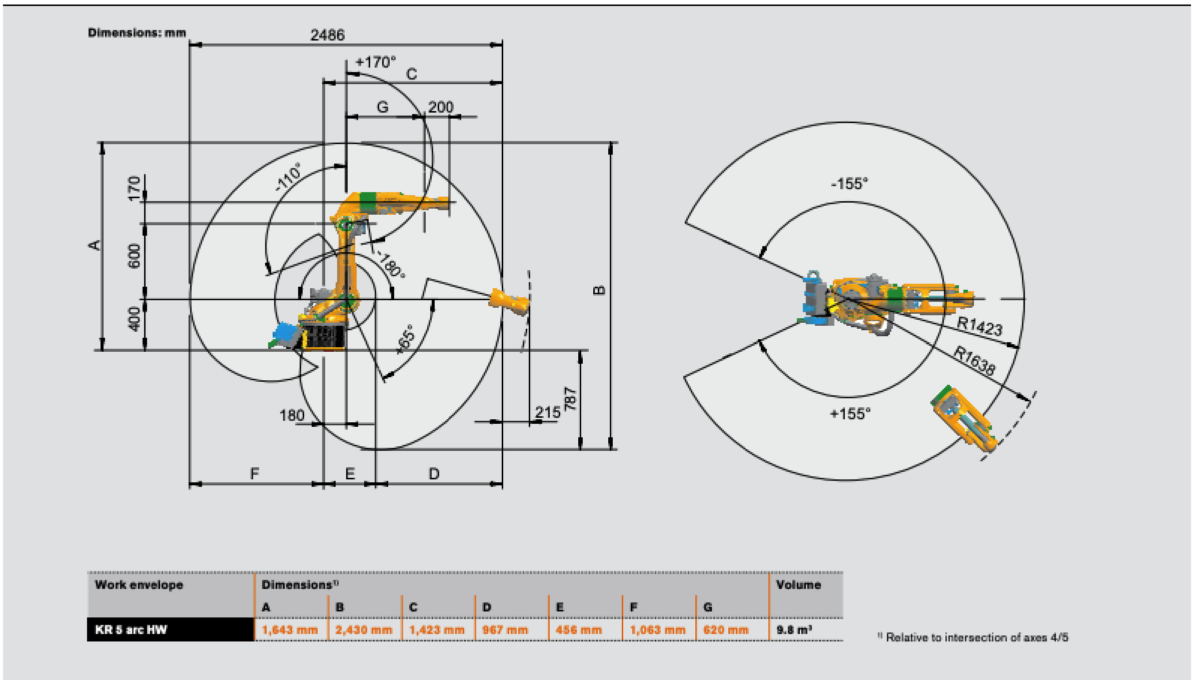
Code snippet of the KR5 Robot [3]

```
Running KUKA simulation...

Displaying qz configuration...
Joint angles: [0. 0. 0. 0. 0. 0.] degrees

Displaying qr configuration...
Joint angles: [45. 60. 45. 30. 45. 30.] degrees

Displaying qk1 configuration...
Joint angles: [45. 60. 45. 30. 45. 30.] degrees

Displaying qk2 configuration...
Joint angles: [45. 60. 30. 60. 45. 30.] degrees

Displaying qk3 configuration...
Joint angles: [30. 60. 30. 60. 30. 60.] degrees

Now showing full trajectory through all configurations...
Moving through all configurations: qz → qr → qk1 → qk2 → qk3
```

Output of the KR5 simulation [3]

Workspace Exploration: KUKA KR5 and CyberKnife®

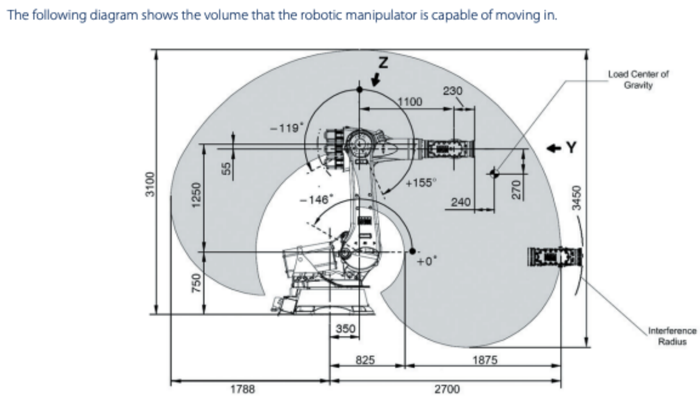
Given the challenges of calculating and plotting the workspace from transformation matrices, and without detailed engineering specifications of joint and link parameters, workspace kinematics will be performed using the **manufacturer-provided workspace diagrams** below.



Work envelope	Dimensions[1]							Volume
	A	B	C	D	E	F	G	
KR 5 arc HW	1,643 mm	2,430 mm	1,423 mm	967 mm	456 mm	1,063 mm	620 mm	9.8 m³

[1] Relative to intersection of axes 4/5

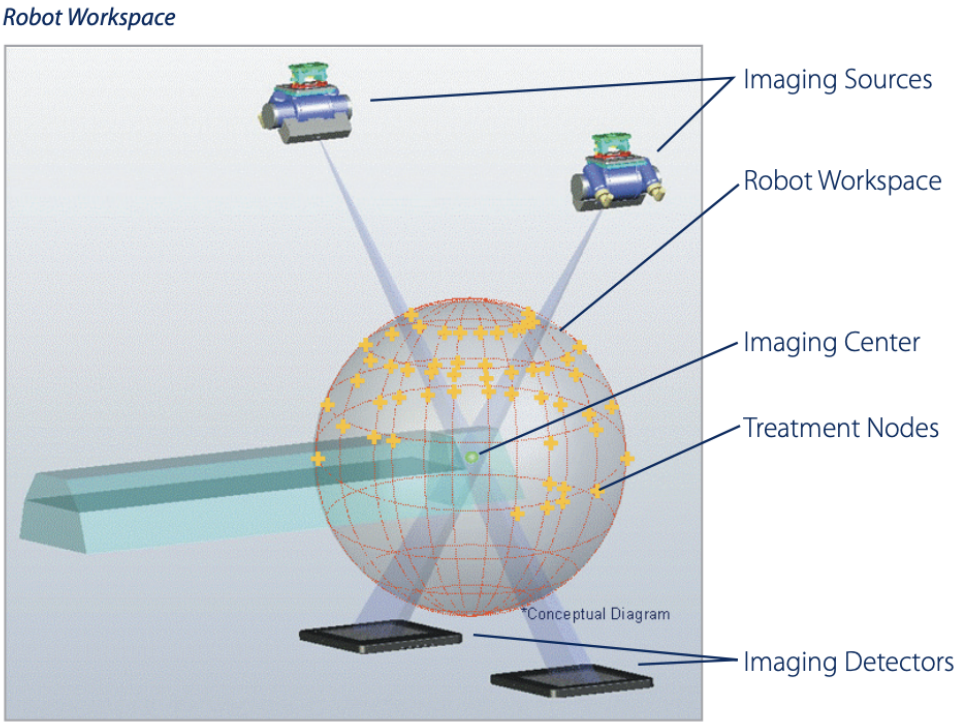
Kuka KR5 workspace specification [9]



CyberKnife® workspace specification [8]

The workspace diagrams from the Kuka KR5 [9] and CyberKnife® [8] engineering specifications shows similar arc-like motion paths and **workspace voids**—areas within the robot's reach that remain inaccessible. Both systems have a maximum reach envelope that defines their operational boundaries. The KR5's envelope has precise dimension lines (1,423–1,643 mm from the base frame), while the CyberKnife® has comparable outer limits (approximately 2,700–3,100 mm from the base frame).

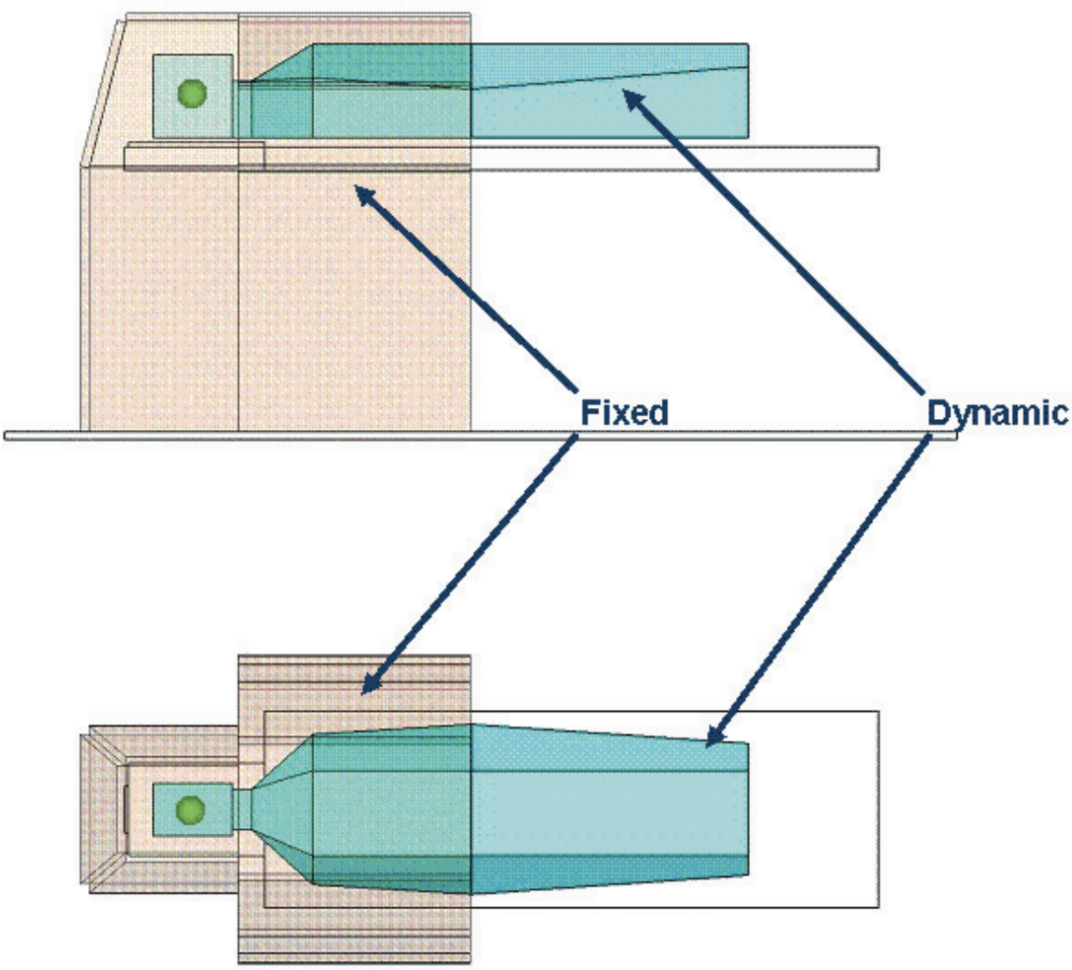
2.2.1 Workspace



The CyberKnife® workspace visualization [5]

In addition, the CyberKnife® system operates within a carefully pre-programmed workspace that ensures safe and precise radiation delivery. This workspace incorporates the physical layout of the treatment suite, including the treatment couch, imaging equipment, and room boundaries. To prevent collisions, the system follows optimized paths between designated nodes - specific points in space where the robot can safely deliver radiation treatment from multiple angles. While the workspace visualization shown above is conceptual, the actual treatment paths are dynamically adapted based on the target location and individual patient anatomy. **This flexibility allows for highly personalized treatment delivery while maintaining strict safety protocols**.

2.2.2 Safety Zones



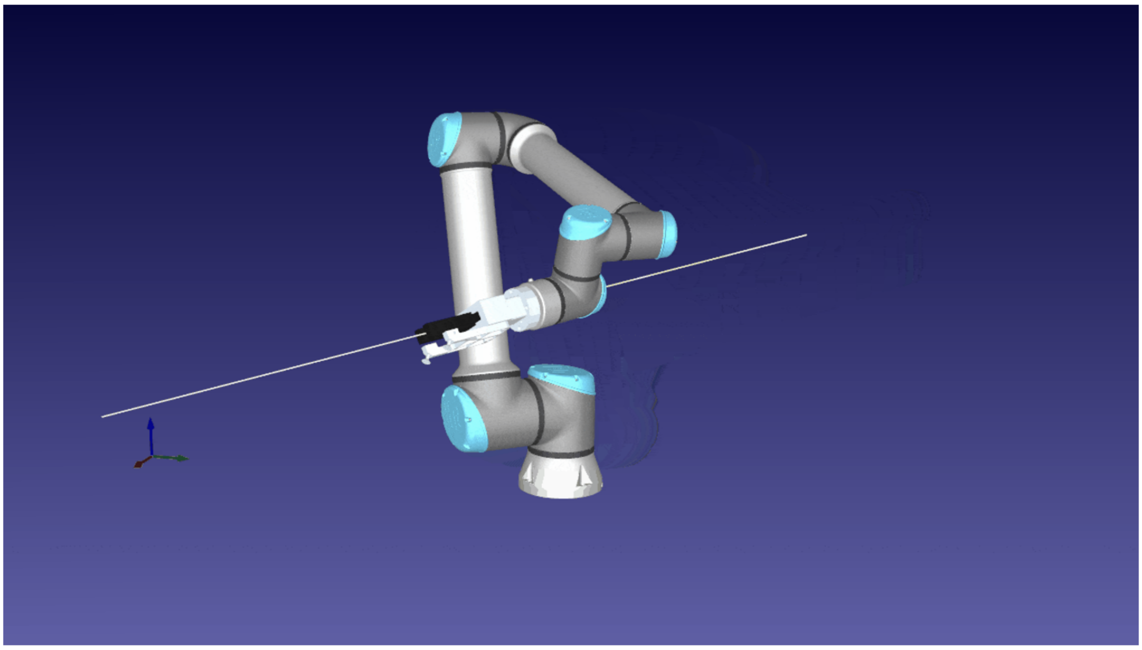
Integrated safety zone systems [5]

Finally, the robot's workspace incorporates sophisticated safety features to prevent patient contact. A dual-layer safety zone system surrounds the patient and treatment couch as shown above. The fixed safety zone remains anchored to the imaging center and treatment area, while the dynamic safety zone adapts to encompass the patient's entire body. For additional protection, a contact detection sensor is installed at the secondary collimator housing on the linear accelerator. If this sensor detects any contact, it triggers an immediate Emergency Stop (E-STOP), instantly halting all system movement.

Quick Notes on Singularities

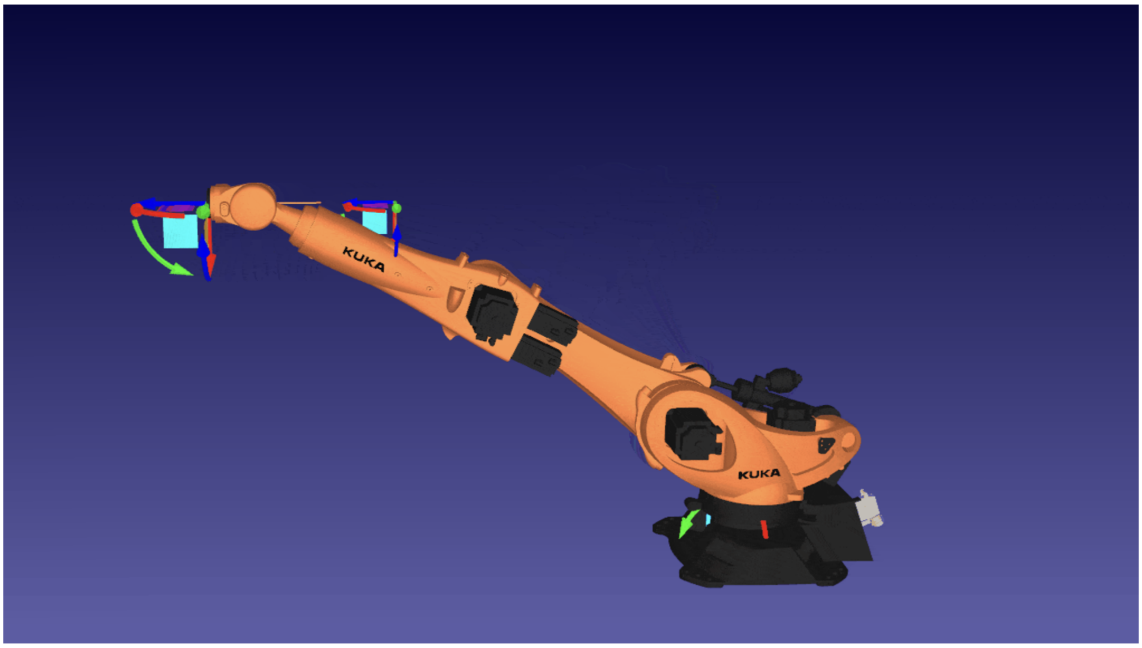
Like other 6 DOF general manipulators, the CyberKnife® can encounter three main types of singularities:

- **Wrist Singularity**: A wrist singularity occurs when the axes of the robot's Joint 4 and Joint 6 become either "coincident" or parallel, depending on the robot. **Coincident lines** are those that are both parallel to each other *and* share a point—in other words, two separate lines merge into one.



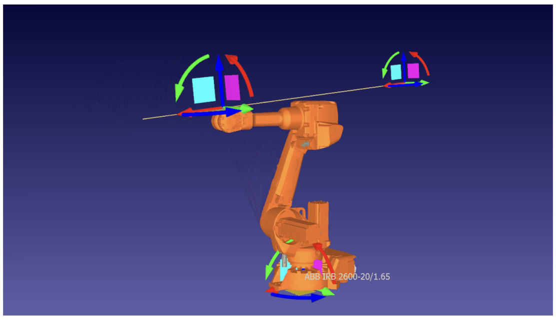
The wrist joints move infinitely fast in the middle of the line as it passes through a wrist singularity [4]

- **Elbow Singularity**: This occurs when the elbow joint (joint 3) reaches full extension or retraction, similar to the PincherX 100's fully extended singularity mentioned above. Put simply, **the robot has "stretched too far."** The control system carefully plans paths to avoid these positions to maintain controllability throughout treatment.

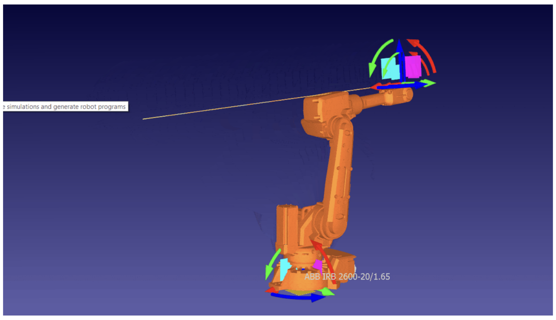


Elbow Singularity by Hill [4]

- **Shoulder Singularity**: This occurs when the target point aligns with Joint 1's rotation axis (directly above or below the robot's base). As the robot approaches a shoulder singularity, the motors in Joints 1 and 4 attempt to rotate 180° at infinite speed, as shown below courtesy of Alex Owen-Hill [4].



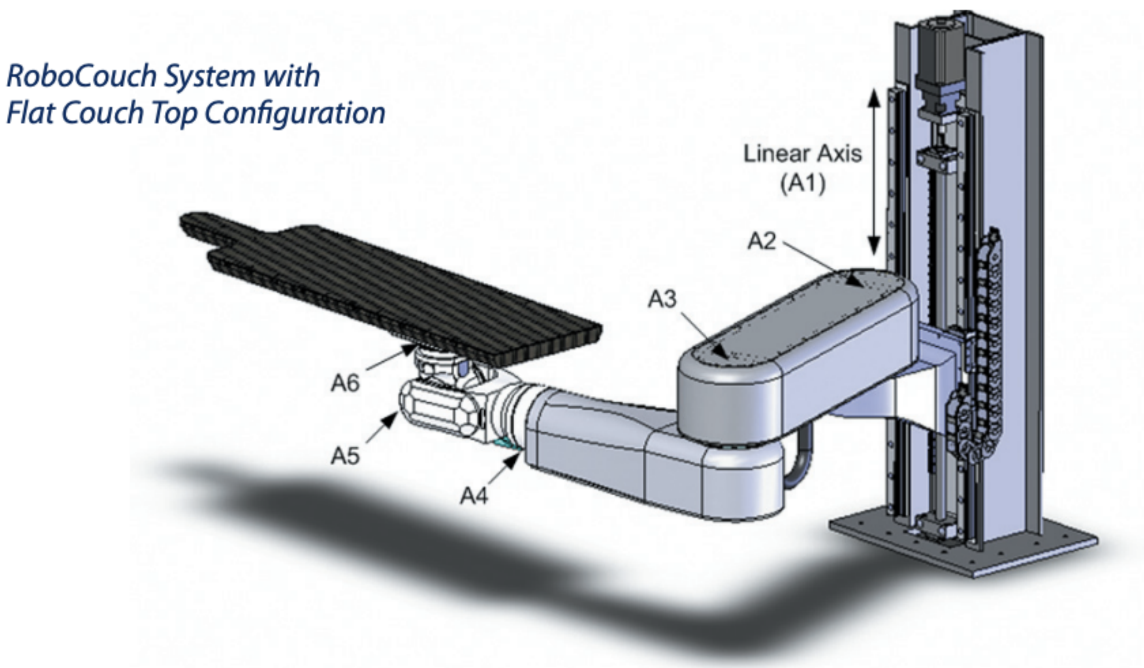
Shoulder Singularity Before by Hill [4]



Shoulder Singularity After by Hill [4]

Given that the CyberKnife® is a mature robot, I believe it uses **preoperative treatment planning** and **workspace analysis** to avoid the singularities mentioned above. Its 6 degrees of freedom allow for redundancy resolution and weighted optimization based on movement goals. For example, the highest priority is maintaining precise beam delivery, followed by avoiding singularities, and then optimizing movement efficiency. As mentioned in the workspace exploration section above, the robot workspace includes cameras for real-time monitoring—if the system detects an approaching singularity, it can dynamically adjust its path.

The RoboCouch® System



The RoboCouch® system configuration [6]

In addition to the robotic arm, the RoboCouch® System enhances the CyberKnife® System's precision through a sophisticated 6-DOF mechanism for patient positioning and radiation delivery. **While it offers six more degrees of freedom, the system incorporates several safety features and movements are deliberately constrained for patient safety and comfort**. The system's five rotational axes and one linear axis (protected by a free-standing cover) are programmed with specific limitations on roll and pitch angles to ensure patient comfort and safety during treatment procedures.

This is effectively a redundant system! This dual-system approach creates several important advantages that enhance both treatment effectiveness and patient experience. From a treatment perspective, this dual-robot system offers **distributed optimization.** The RoboCouch® handles the broader, slower patient positioning movements, while the CyberKnife® robot manages the precise, rapid movements needed for radiation delivery. This arrangement creates remarkable treatment flexibility. Instead of needing to move a patient to access different treatment angles, the CyberKnife® robot can move around the patient while the RoboCouch® makes subtle adjustments to maintain optimal positioning.

From a technical perspective, the dual system offers **error compensation**. If one system has slight positioning inaccuracies, the other can compensate. For example, if the patient moves slightly on the RoboCouch®, the CyberKnife® robot can adjust its targeting in real-time rather than needing to reposition the entire patient. Treatment planning also becomes more flexible. Medical teams can choose the optimal combination of patient position and robot movements to avoid potential collisions and minimize treatment time to ensure a good patient experience.

Medical Application

#CyberKnife® System Treatment for Multiple Tumor Types

For more information about the CCyberKnife®system head to our patient website: https://cCyberKnife®com

https://www.youtube.com/watch?time_continue=2&v=tS3FD11OfMc&embeds_referring_euri=https%3A%2F%2Fcyberknife.com%2F&embeds_referring_origin=https%3A%2F%2Fcyberknife.com&source_ve_path=MjM4NTE

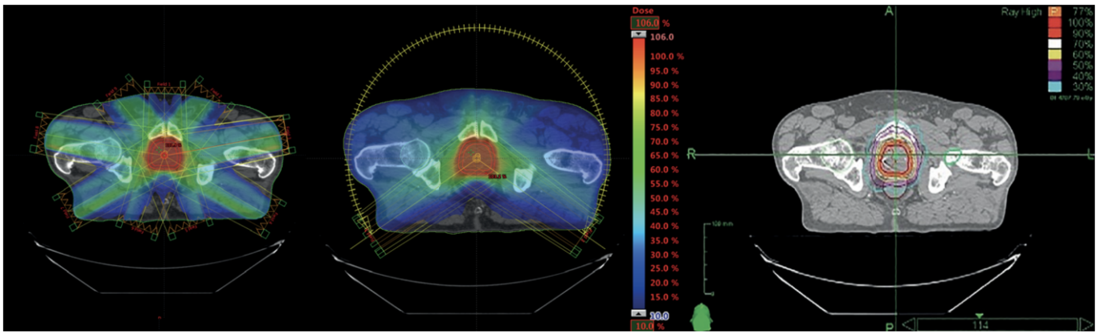
One of the examples given in the CyberKnife® treatment video is treating prostate cancer. This case is particularly interesting because the prostate moves as the bladder and rectum fill. The CyberKnife® can track this motion and adjust its radiation beam to maintain precise targeting. This advanced kinematics allows for smaller margins of error and helps avoid excessive radiation exposure. They've mentioned that the CyberKnife® reduces the treatment course to 5 fractions, as opposed to 42!

> *“Your prostate is actually moving as we’re treating. Radiation oncologists hate missing the tumors! So what do we do? We add an extra margin around the prostate so we don’t miss the tumor. Now, that means i’m getting a little bit more radiation to your bladder, your rectum and your penile bulb. All areas that I don’t want to damage.”*

According to the video, the CyberKnife® system is highly effective for treating various tumor types, offering exceptional precision with up to 1mm accuracy and controlled radiation dose fall-off to protect surrounding tissues. This precision is particularly valuable when treating sensitive areas - for example, protecting the duodenum during pancreatic cancer treatment, or minimizing damage to healthy tissue during lung cancer treatment. The system can even adjust for natural body movements, such as breathing, ensuring accurate tumor targeting throughout the procedure.

Finally, the video claims that the CyberKnife® system offers significant cost advantages by reducing the number of treatments from 42 with IMRT to just 5. Patients show consistent decreases in PSA (prostate-specific antigen) levels, with reduced risk of long-term complications in their later years.

A research article titled "*Does CyberKnife® improve dose distribution versus IMRT and VMAT on a linear accelerator in low-risk prostate cancer?*" from the Department of Medical Physics and Electroradiology of Greater Poland Cancer Center [7] supports these claims. The study's dosimetric results demonstrate the CyberKnife®'s superior precision, **showing a better conformity index (1.07) compared to IMRT (1.25) and VMAT (1.17).** This indicates more precise dose delivery to the target area, supporting the system's high-precision capabilities.



Example dose distributions on a transversal scan for (A) IMRT plan, (B) VMAT plan and (C) CyberKnife® plan for the same patient [7]

The paper also validates the benefits of hypo fractionation, though with slightly different parameters. **While the video describes a reduction from 42 to 5 fractions, the study implemented a protocol of 36.25 Gy delivered in 5 fractions (7.25 Gy per fraction)**. This still demonstrates a significant reduction from conventional treatment courses, confirming the efficiency claims of the CyberKnife®.

Conclusion ✨

We can easily see the stark contrast between simple robotic manipulators and specialized medical robots in their kinematic capabilities and workspace characteristics. The PincherX 100's 4-DOF architecture, while effective for auxiliary medical tasks like equipment handling and rehabilitation exercises, **proves unsuitable for complex surgical procedures** due to its limited tool orientation, restricted dextrous workspace, and inability to maintain consistent control through complex trajectories. In contrast, while the CyberKnife® system's 6-DOF design **potentially introduces more singularities**, its integration with the sophisticated RoboCouch® patient positioning system demonstrates how **advanced kinematics can be strategically combined to enable highly precise medical procedures**.

The differences in workspace volume, singularity management, and end-effector control highlight why medical robots require specialized kinematic configurations. After all, the success of medical robotics applications relies on the thoughtful integration of kinematic design principles with specific clinical requirements and patient comfort.

References 📚

Simulation Resources

1. **PincherX-100 Documentation** by *Interbotix*

> PincherX-100 — Interbotix X-Series Arms Documentation
> The PincherX-100 Robot Arm belongs to the Interbotix X-Series family of arms featuring the DYNAMIXEL X-Series Smart Servo Motors. The X-Series actuators offer high torque, efficient heat dissipation and great durability all at a smaller form factor over previous DYNAMIXEL servos. The
> https://docs.trossenrobotics.com/interbotix_xsarms_docs/specifications/px100.html

2. **PincherX 100 Description Package** by *Cristian Chitiva* from the Universidad Nacional de Colombia.

> https://github.com/cychitivav/px100_description

3. **Kinematics Simulation and Fusion Model** by *Frank Lü from the* University of Surrey

> https://github.com/frankcholula/mr

> Fusion
> Share 2D and 3D design files and project files with anyone.
> https://a360.co/4ibQkio

Core Concepts and Learning Resources

3. **Denavit-Hartenberg Parameters Visualization** by *Addison Sears-Collins*

> How to Find Denavit-Hartenberg Parameter Tables
> In this section, we'll learn how to find the Denavit-Hartenberg Parameter table for robotic arms. This method is a shortcut for finding homogeneous transformation matrices and is commonly seen in
> https://automaticaddison.com/how-to-find-denavit-hartenberg-parameter-tables/

Addison kindly shared his tutorial on **Coding DH Tables Using Python** after I reached out to him on LinkedIn. You can find the tutorial below.

Coding Denavit-Hartenberg Tables Using Python

Now that we've seen how to fill in Denavit-Hartenberg Tables, let's see how we can use these tables and Python code to calculate homogeneous transformation matrices for robotic arms. Our goal is to

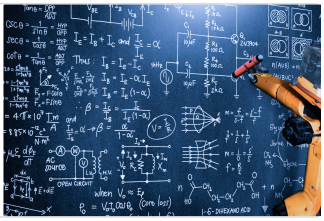
https://automaticaddison.com/coding-denavit-hartenberg-tables-using-python/

4. **Robot Singularities: What Are They and How to Beat Them** by Alex Owen-hill from RoboDK

Robot Singularities: What Are They and How to Beat Them - RoboDK blog

Robot singularities can cause havoc when you are programming a robot. But, what is a singularity? Here's how to handle singularities...

https://robodk.com/blog/robot-singularities



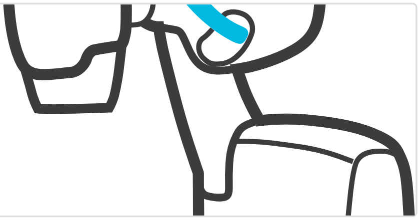
Robot Specifications

5. **The CyberKnife® System** by Accuray

Home - CyberKnife®

Accuray's CCyberKnife®System provides true robotic precision for personalized radiotherapy. Learn about CCyberKnife® find a treatment center and more.

https://cyberknife.com/



CyberKnife® Equipment Specifications by Accuray

www.CyberKnife®latin.com

https://www.cyberknifelatin.com/pdf/brochure-tecnico.pdf

6. **Kuka KR5 Hardware Specification** by Kuka

www.kuka.com

https://www.kuka.com/-/media/kuka-downloads/imported/8350ff3ca11642998dbdc81dcc2ed44c/pf0012_kr_5_arc_hw_en.pdf

Kuka KR Quantec by Kuka

KR QUANTEC | KUKA AG

All-purpose robot series with the largest payload and range in the high payload class for any application from automotive, to foundry, to medical.

https://www.kuka.com/en-gb/products/robotics-systems/industrial-robots/kr-quantec

Medical Papers

7. **Does CyberKnife® improve dose distribution versus IMRT and VMAT on a linear accelerator in low-risk prostate cancer?** by *Dorota Maria Borowicz*, *Agnieszka Skrobała*, *Marta Kruszyna-Mochalska*, and *Julian Malicki* from the Department of Medical Physics and Electroradiology Department, University of Medical Sciences, Poznan, Poland

   pmc.ncbi.nlm.nih.gov
   https://pmc.ncbi.nlm.nih.gov/articles/PMC9122296/pdf/raon-56-259.pdf

Other Images

8. **Standard Crawler Excavators PC950LC-11** by Komatsu

   Komatsu Standard Crawler Excavator PC950LC-11
   Komatsu PC950LC-11 comply with the latest industry standards and work in synergy to minimise risks to people in and around the machine. EU Stage V engine Independent swing system Komatsu fuel-saving
   https://www.komatsu.eu/en/excavators/crawler-excavators/standard-crawler-excavators/pc950lc-11

9. **A Planar Parallel Device for Neurorehabilitation** by MDPI

   A Planar Parallel Device for Neurorehabilitation
   The patient population needing physical rehabilitation in the upper extremity is constantly increasing. Robotic devices have the potential to address this problem, however most of the rehabilitation robots are
   https://www.mdpi.com/2218-6581/9/4/104

   robotics