Transforms:

world-frame = robots origin
tag4-frame = fixed offset from world frame
tag14.img1-frame = tag14 in image 1 (only tag14)
tag14.img2-frame = tag14 in image 2 (both tags)

To triangulate entry point:

P1 = projection matrix for image 1
P2 = projection matrix for image 2
k = camera matrix

→ P1 = k x T_img1.camera_world

→ T_img1_camera-world =
T_img1camera_tag14 x T_tag14-img2camera x T_img2camera_tag4 x T_tag4-world

→ P2 = k x T_img2-camera_world

→ T_img2camera_world = T_img2camera_tag4 x T_tag4-world

returns 3D point in world frame

Triangulate target point:

P_top = K_top_carm × T_topcam_tag14

T_topcam_tag14 =
T_topcam_toptag × Ttoptag-tag14

P_side = K_side_carm × T_sidecam_ tag14

T_sidecam_tag14 = T_sidecam_sidetag × Tsidetag-toptag
× T_toptag-tag14

returns 3D point in tag14 frame (P_tag14)

P_img2cam = T_img2cam_tag14 × P_tag14
P_tag4 = T_tag4-img2cam × P_img2cam
P_world = T_world_tag4 × P_tag4

Trajectory = Target point - entry point

calculate rpy as normal
calculate distance w/ euclidean distance