Stiffness model for Tripteron robot

1 Virtual Joints model

- Consider each of three legs separately. The model for them are same the only difference is axis of rotations and translations of links and joints.
- Inverse kinematics for one leg:
  - z coordinate defined by a prismatic joint position $q_1 =$ z
  - q_2 and q_3 can be found as inverse kinematics for planar 2d manipulator:
    $q_3 = \arccos(\frac{x^2+y^2-L_1^2-L_2^2}{2L_1L_2})$
    $q2 = \arctan(\frac{y}{x}) - \arctan(\frac{L_2\sin(q_3)}{L_1+L_2\cos(q_3)})$
- For every leg there are 4 virtual joints. One is 1d (after active prismatic joint) and 3 6d (after each link).
- So the direct kinematics for this robot can be written as follows:
  $T = T_z(q_1)T_z(\theta_1)T_{6d}(\theta_2)R_z(q_2)T_x(L_1)T_{6d}(\theta_3)R_z(q_3)T_x(L_2)T_{6d}(\theta_4)R_z(q_4)$
- After that we construct K_θ matrix base on arbitrarily chosen values of stiffness properties of links and active joints
- Then build deflection map for the robot by measurement end-effector displacement in some set of points x, y (z chose as constant since end-effector move on one plane):
  1. For each point compute inverse kinematics assuming that virtual joints positions are 0 (since they are small)
  2. Compute Jacobians by virtual joints positions J_θ and passive joints J_q
  3. Compute stiffness matrix K_C
  4. Compute end-effector displacement: $\Delta t = K_C^{-1}W$
- Deflection map plots:

2 Github link: