(a) Derive the velocity kinematic map J(d1, θ 2) for the PR robot.

From last project we found the A(q) matrix as follows:

$$A(d1, \theta 2\,) \;=\; A1.A2 \;=\; \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2\sin(\theta_2) \\ 0 & 0 & 1 & l_1 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then Jacobian can be computed as follows:

$$O_2^0 = \begin{bmatrix} l_2\cos(\theta_2) \\ l_2\sin(\theta_2) \\ l_1 + d_1 \end{bmatrix}$$

$$V_2^0 = \begin{bmatrix} \dfrac{\partial O_2^0}{\partial d_1} & \dfrac{\partial O_2^0}{\partial \theta_2} \end{bmatrix} x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix} = J_v \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$V_2^0 = \begin{bmatrix} 0 & -l_2\sin(\theta_2) \\ 0 & l_2\cos(\theta_2) \\ 1 & 0 \end{bmatrix} x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix} = J_v \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$w_2^0 = [\rho_1 k \quad \rho_2 R_1^0 k] \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix} = J_w \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$w_2^0 = [\rho_1 k \quad \rho_2 I k] \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix} = J_w \; x \begin{bmatrix} \dot{d}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

$$J(d1, \theta 2) = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} 0 & -l_2\sin(\theta_2) \\ 0 & l_2\cos(\theta_2) \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

(b) First consider a random initial location and position of the robot. What is the corresponding Jacobian matrix? Compare it with that manually computed.

At the beginning of my program, I have set the joint values to $d_1 = -0.1, \theta_2 = -1$ *and* $l_2 = 1$

Corresponding Jacobian is $J = \begin{bmatrix} 0 & -l_2 \sin(\theta_2) \\ 0 & l_2 \cos(\theta_2) \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -0.84 \\ 0 & 0.54 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$

This result is the same result with the printed result that is seen on the console as First Jacobian:...

This Jacobian is printed first so you need to check before it gets lost.