Jacobians – Velocity Transformation

Craig Carignan
Glen Henshaw

September 23, 2025

Abstract

Jacobian (Ref Sec. 5.7; Exercises 5.17, SS 3.1, SS 3.2)

0.1 Jacobians

The "Jacobian" is actually a "Jacobian matrix" which relates the differentials of one coordinate set to another:

$$y_1 = f_1(x_1, x_2) \tag{1}$$

$$y_2 = f_2(x_1, x_2) \tag{2}$$

or

$$\underline{y} = \underline{f}(\underline{x}) \tag{3}$$

Taking partial derivatives of both sides, we get

$$\delta\underline{y} = \frac{\partial \underline{f}}{\partial \underline{x}} \delta\underline{x}$$

$$\frac{\partial \underline{f}}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

or

$$\dot{\underline{y}} = \frac{\partial \underline{f}}{\partial \underline{x}} \dot{\underline{x}} \equiv J \dot{\underline{x}} \tag{4}$$

Note that $J \equiv J(\underline{x})$ if $\underline{f}(\underline{x})$ is nonlinear.

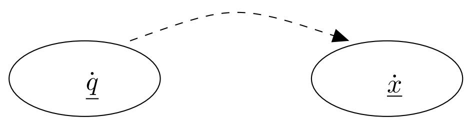
In robotics, the two coordinate sets are typically the joint angles and the end effector pose (position and orientation):

$$\underbrace{\begin{bmatrix} {}^i\underline{\mathcal{V}} \\ {}^i\underline{\omega} \end{bmatrix}}_{\substack{\text{tool} \\ \text{velocity} \\ [M]}} = \underbrace{\begin{bmatrix} {}^iJ_{tran} \\ {}^iJ_{rot} \end{bmatrix}}_{\substack{\text{Jacobian} \\ J \\ [M \times N]}} \underbrace{\dot{\underline{q}}}_{\substack{\text{joint} \\ \text{rates} \\ [N]}} \tag{5}$$

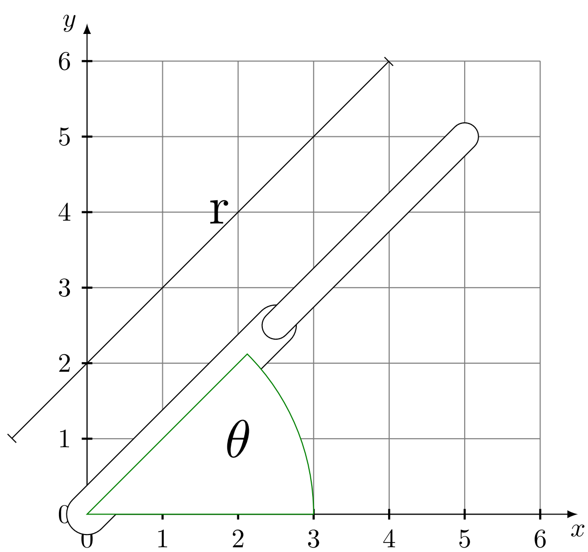
where

$$\underline{\dot{q}}_i = \begin{cases} \underline{\dot{\theta}}_i & \text{revolute} \\ \underline{\dot{d}}_i & \text{prismatic} \end{cases}$$

So the Jacobian J transforms vectors of joint rates to vectors of end effector rates:



0.2 EXAMPLE: Polar robot translation Jacobian



Let's first write the forward kinematics for this arm. From inspection,

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \end{aligned}$$

or

$$\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\underline{p}} = \underbrace{\begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}}_{\underline{f}(r,\theta)}$$

or

$$\underline{p} = \underline{f}(\underline{q})$$

where

$$\begin{aligned} \underline{p} &= [x \;\; y]^t \\ \underline{q} &= [r \;\; \theta]^t \end{aligned}$$

then

$$\dot{\underline{p}} = J_{tran}(\underline{q}) \Rightarrow J_{tran}(\underline{q}) \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix} \tag{6}$$

This is the "direct differentiation" method. Works okay for arms with few degrees of freedom. Completely intractable for most arms.

Okay, what about rotation?

To start, let's think about a vector ${}^B\underline{p}$ that is constant in frame B. But let's suppose that frame B is itself rotating with respect to an inertially fixed frame A. In other words, the time derivative of the rotation matrix ${}^A_B\dot{R} \neq 0$. Then,

$${}^A\dot{\underline{p}} = {}^A_B\dot{R}\; {}^B\underline{p}$$

or

$${}^A\underline{v}_B = {}^A_B\dot{R}\; {}^B\underline{p}$$

Substituting for ${}^B\underline{p}$, we can get

$${}^A\underline{v}_B = {}^A_B\dot{R}\; {}^A_BR^{-1}\; {}^A\underline{p}$$

We know from equation 5.19 from the text that

$$S_\times = \dot{R}R^{-1}$$

where $S_\times$ is a skew-symmetric matrix. So we can write this as

$${}^A\underline{v}_B = {}^A_BS\; {}^A\underline{p}$$

and if we define

$$\Omega_\times \triangleq \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

and we define the 3–vector

$$\underline{\omega} \triangleq \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

we can verify that

$$\Omega_\times \underline{p} = \omega \times \underline{p}$$

so we can write

$${}^A\underline{v}_B = {}^A\Omega_{\times,B}\; {}^A\underline{p}$$

This means that $\underline{\omega}$ is the angular velocity vector of frame B with respect to frame A.

Now,

$$\dot{R} = \sum_{j=1}^{N} \frac{\partial R(\underline{q})}{\partial q_j} \dot{q}_j$$

"can be rewritten as"

$$\underline{\omega} = J_{rot} \dot{\underline{q}}$$

in the following manner:

$$\Omega_{\times} = R^T(\underline{q}) \sum_{j=1}^{N} \frac{\partial R(\underline{q})}{\partial q_j} \dot{q}_j$$

and we can carry this calculation out for the example above:

$$R(\underline{q}) = \begin{bmatrix} c_\theta & s_\theta & 0 \\ -s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so

$$\dot{R}(\underline{q}) = \begin{bmatrix} -s_\theta & s_\theta & 0 \\ -c_\theta & -s_\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and

$$R \sum_{j=1}^{N} \frac{\partial R(\underline{q})}{\partial q_j} = \begin{bmatrix} 0 & c_\theta^2 - s_\theta^2 & 0 \\ s_\theta^2 - c_\theta^2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

because all of the partials with respect to r are zero. We can then "pick off" the terms in the matrix representing each of ω_x, ω_y, and ω_z. In this case, of course, $\omega_x = 0$ and $\omega_y = 0$ and we are simply left with

$$\omega_z = (s_\theta^2 - c_\theta^2)\dot{\theta}$$

so

$$J_{\text{rot}} = \begin{bmatrix} 0 & s_\theta^2 - c_\theta^2 \end{bmatrix}$$

and the entire Jacobian is generally written as

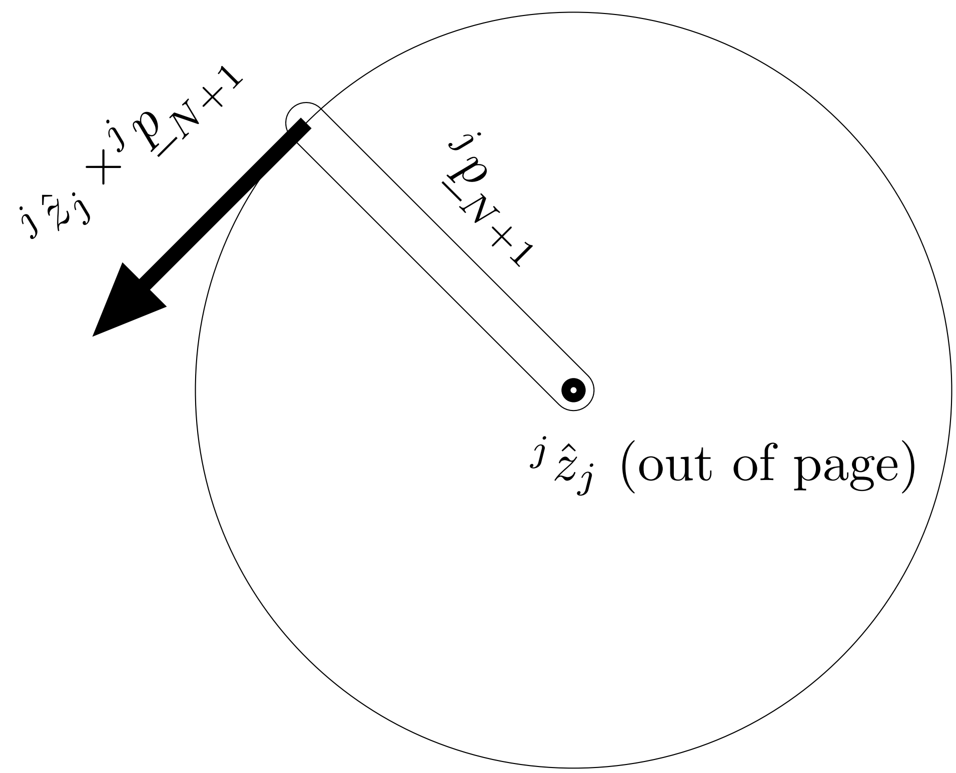
$$J = \begin{bmatrix} c_\theta & -r\ s_\theta \\ s_\theta & -r\ c_\theta \\ 0 & s_\theta^2 - c_\theta^2 \end{bmatrix}$$

See Exercise 5.16 and the RPR Wrist Jacobian handout.

This is the "Direct Differentiation" method for the rotation Jacobian, and it is in general very tedious.

0.3 Cross–product method

There is an alternative method called the "cross–product method" which is more computationally efficient and derives from the velocity propagation method. It's based on the insight that each element of the Jacobian describes the instantaneous motion of the end effector along some direction in terms of the motion of each joint. Graphically:



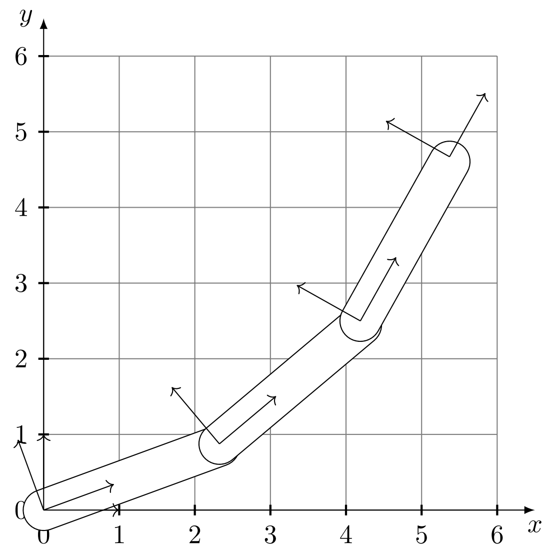
where the circle is the set of points swept out by the end effector and the solid arrow shows the current velocity of the end effector, which of course is normal to the circle. Finding the direction of this vector is, of course, a simple matter of crossing the unit vector passing through the link's center of rotation (out of the plane of the page) with the vector from the center of rotation to the end effector.

Formally, let

$$ {}^i\hat{z}_j \triangleq {}^i({}^j\hat{z}_j) $$

i.e. the third column of ${}^i_jR = {}^i_jR\,{}^jz_j$. Then,

$$ {}^iJ_{rot} = \left[\,{}^i\hat{z}_1, {}^i\hat{z}_2, \cdots, {}^i\hat{z}_N\right]\Lambda $$



where

$$\Lambda \triangleq \mathrm{diag}(\lambda_i) \text{ with } \lambda_i = \begin{cases} 1 & \text{revolute} \\ 0 & \text{prismatic} \end{cases}$$

and

$$\begin{aligned} {}^iJ_{trans} = & \left[{}^i\hat{z}_1 \times \ {}^i({}^1\underline{p}_{N+1}),\, {}^i\hat{z}_2 \times \ {}^i({}^2\underline{p}_{N+1}),\, {}^i\hat{z}_3 \times \ {}^i({}^3\underline{p}_{N+1}) \right] \Lambda \\ & + (I_{N\times N} - \Lambda) \left[{}^i\hat{z}_1,\, {}^i\hat{z}_2, \cdots, {}^i\hat{z}_N \right] \end{aligned}$$

0.4 Changing the frame of the Jacobian

Given ${}^BJ$, find ${}^AJ$.

By definition,

$$\begin{bmatrix} {}^B\underline{\mathcal{V}}_p \\ {}^B\underline{\omega} \end{bmatrix} =^B J(\underline{q})\underline{\dot{q}}$$

Velocity is a free vector, so:

$$\left. \begin{array}{rcl} {}^A\underline{\mathcal{V}}_p & = & {}^A_BR\, {}^B\underline{\mathcal{V}}_p \\ {}^A\underline{\omega} & = & {}^A_BR\, {}^B\underline{\omega} \end{array} \right\} \Rightarrow \begin{bmatrix} {}^A\underline{\mathcal{V}}_p \\ {}^A\underline{\omega} \end{bmatrix} = \begin{bmatrix} {}^A_BR & 0_{3\times3} \\ 0_{3\times3} & {}^A_BR \end{bmatrix}^B J(\underline{q})\underline{\dot{q}})$$

0.5 Example: 3–link planar manipulator

0.5.1 A: Find ${}^iJ_{rot} \Rightarrow {}^0_1R\,{}^1_2R$

$${}^0_1R = \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ {}^0_2R = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix},\ {}^0_3R = \begin{bmatrix} c_{123} & -s_{123} & 0 \\ s_{123} & c_{123} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$
{}^0J_{rot} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = {}^2J_{rot} \Rightarrow {}^0\underline{\omega}_3 = {}^2\underline{\omega}_3 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}
$$

0.5.2 B: Find ${}^iJ_{trans}$

(1) Use differentiation method

$$
{}^0\underline{p}_4 = \begin{bmatrix} l_1c_1 + l_2c_{12} + l_3c_{123} \\ l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 \end{bmatrix}
$$

Factoring,

$$
{}^0j_{trans} = \frac{\partial\, {}^0\underline{p}_4}{\partial \underline{\theta}} = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} & -l_2s_{12} - l_3s_{123} & -l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} & l_2c_{12} + l_3c_{123} & l_3c_{123} \\ 0 & 0 & 0 \end{bmatrix}
$$

$$
{}^2J_{tran} = {}^2_0R\,{}^0J = \begin{bmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} & -l_2s_{12} - l_3s_{123} & -l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} & l_2c_{12} + l_3c_{123} & l_3c_{123} \\ 0 & 0 & 0 \end{bmatrix}
$$

or

$$
= \begin{bmatrix} l_1s_2 - l_3s_3 & -l_3s_3 & -l_3s_3 \\ l_1c_2 + l_2 + l_3c_3 & l_2 + l_3c_3 & l_3c_3 \\ 0 & 0 & 0 \end{bmatrix}
$$

Notice that ${}^2J_{tran}$ is much simpler than ${}^0J_{tran}$.

(2) Use "Cross Product" method

$$
{}^0J_{tran} = \begin{bmatrix} {}^0\hat{z}_1 \times {}^0({}^1\underline{p}_4) & {}^0\hat{z}_2 \times {}^0({}^2\underline{p}_4) & {}^0\hat{z}_3 \times {}^0({}^3\underline{p}_4) \end{bmatrix}
$$

$$
{}^0\hat{z}_1 = {}^0\hat{z}_2 = {}^0\hat{z}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

$$
{}^3\underline{p}_4 = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}
$$

$$
{}^2\underline{p}_3 = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}
$$

$$
{}^1\underline{p}_2 = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}
$$

so

$$
{}^0({}^3\underline{p}_4) = \begin{bmatrix} l_3c_{123} \\ l_3s_{123} \\ 0 \end{bmatrix}
$$

$$
{}^0({}^2\underline{p}_4) = {}^0({}^2\underline{p}_3) + {}^0({}^3\underline{p}_4) = \begin{bmatrix} l_2c_{12} + l_3c_{123} \\ l_2s_{12} + l_3s_{123} \\ 0 \end{bmatrix}
$$

$$
{}^0({}^1\underline{p}_4) = {}^0({}^1\underline{p}_2) + {}^0({}^2\underline{p}_4) = \begin{bmatrix} l_1c_1 + l_2c_{12} + l_3c_{123} \\ l_1s_1 + l_2s_{12} + l_3s_{123} \\ 0 \end{bmatrix}
$$

Note that these two equations can be obtained recursively. Performing the cross product yields the same result for ${}^0J_{trans}$:

$$
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times^0 ({}^1\underline{p}_4) = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} \\ 0 \end{bmatrix}
$$

$$
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times^0 ({}^2\underline{p}_4) = \begin{bmatrix} -l_2s_{12} - l_3s_{123} \\ l_2c_{12} + l_3c_{123} \\ 0 \end{bmatrix}
$$

and

$$
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times^0 ({}^3\underline{p}_4) = \begin{bmatrix} -l_3s_{123} \\ l_3c_{123} \\ 0 \end{bmatrix}
$$

Assembling these vectors into the Jacobian matrix gives the same result as with the direct differentiation method:

$$
{}^0j_{trans} = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} & -l_2s_{12} - l_3s_{123} & -l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} & l_2c_{12} + l_3c_{123} & l_3c_{123} \\ 0 & 0 & 0 \end{bmatrix}
$$

To compute ${}^2J_{trans}$, we do the same calculations, but in frame 2:

$$
{}^2J_{trans} = \begin{bmatrix} {}^2\hat{z}_1 \times {}^2({}^1\underline{p}_4) & {}^2\hat{z}_2 \times {}^2({}^2\underline{p}_4) & {}^2\hat{z}_3 \times {}^2({}^3\underline{p}_4) \end{bmatrix}
$$

In our example, ${}^2\hat{z}_1 = {}^2\hat{z}_2 =^2 \hat{z}_3 = [0\ 0\ 1]^T$, so starting with the last term:

$$
{}^2({}^3\underline{p}_4) = \begin{bmatrix} l_3c_3 \\ l_3s_3 \\ 0 \end{bmatrix}
$$

then

$$
{}^2({}^2\underline{p}_4) = {}^2({}^2\underline{p}_3) + {}^2({}^3\underline{p}_4) = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_3c_3 \\ l_3s_3 \\ 0 \end{bmatrix}
$$

and finally

$$^2({}^1\underline{p}_4) = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_2 + l_3 c_3 \\ l_3 s_3 \\ 0 \end{bmatrix} = \begin{bmatrix} l_2 + l_2 c_2 + l_3 c_3 \\ -l_2 s_2 + l_3 s_3 \\ 0 \end{bmatrix}$$

Performing the cross product yields the same results for ${}^2J_{trans}$. Notice how much easier it is to compute using this method!