# Jacobians in the Force Domain

Craig Carignan
Glen Henshaw

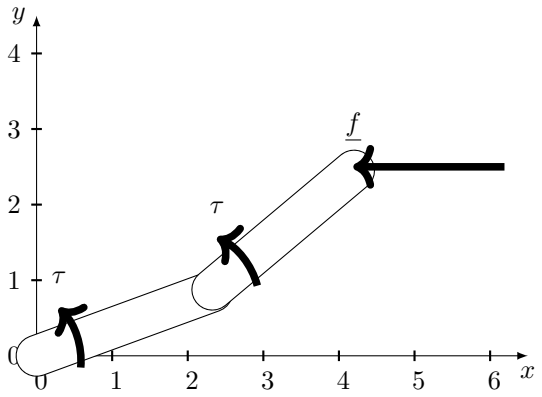
October 8, 2023

**Abstract**

## 1 Intro

Let's consider moment and force propagation through a link, instead of angular and linear velocity propagation.

In the static case:



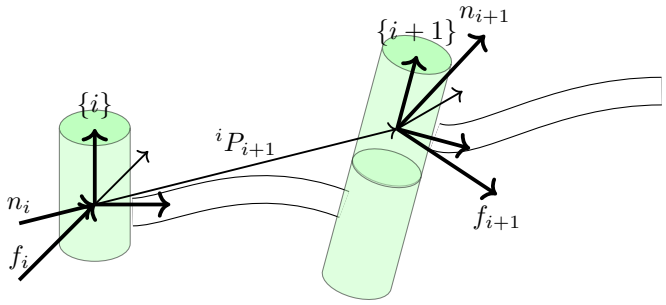

What $\underline{\tau}$ is required to counteract $\underline{f}$? Since this is a statics analysis, we'll treat the robot as a structure, locking all of the joints.

Some definitions:

$\underline{f}_i$ force exerted on link i by link i-1
$\underline{n}_i$ torque exerted in link i by link i-1

Since this is a static analysis, we assume that the forces and the torques must add to zero. Specifically,

$$^i f_i - {}^i f_{i+1} = 0$$




and

$$ {}^in_i - {}^in_{i+1} - {}^iP_{i+1} \times {}^if_{i+1} = 0 $$

In order to recursively calculate the forces and torques exerted on each link, we have to start with the last link and work backwards, towards the base frame. And we need equations that are formulated entirely in terms of the forces and torques exerted by the distal link. These can easily be seen to be

$$ {}^if_i = {}^if_{i+1} \quad (1) $$

$$ {}^in_i = {}^in_{i+1} + {}^iP_{i+1} \times {}^if_{i+1} \quad (2) $$

We'd really like to express the forces and torques entirely in their "own" frames, so we'll rotate them as such:

$$ {}^if_i = {}^i_{i+1}R\,{}^{i+1}f_{i+1} \quad (3) $$

$$ {}^in_i = {}^i_{i+1}R\,{}^{i+1}n_{i+1} + {}^iP_{i+1} \times {}^if_{i+1} \quad (4) $$

NOTE that all forces and moment vectors are absorbed by the structure of the arm except for those exerted about the axis of rotation of the joint. So to balance the force/moment equation, we have to isolate the $z$–component. If the joint is revolute, this is $\tau_i = {}^in_i^T\,{}^i\hat{Z}_i$. If the joint is prismatic, this is $\tau_i = {}^if_i^T\,{}^i\hat{Z}_i$.

## 1.1 Virtual Work

The concept here is that work can be expressed in terms of either motion of the joints or motion of the end effector, and that these two terms must be equal. Work is the inner product of a vector force or torque with a vector displacement. In the case of our diagram, this is

$$ \underbrace{\mathcal{F} \cdot \delta\mathcal{X}}_{\substack{\text{Cartesian}\\ \text{space}}} = \underbrace{\tau \cdot \delta\Theta}_{\substack{\text{Joint}\\ \text{space}}} $$

where $\mathcal{F}$ is a Cartesian force–moment vector acting on the end effector, $\delta\Theta$ is a infinitesimal Cartesian displacement of the end effector, $\tau$ is the vector of torques at the joints, and $\delta\Theta$ is an infinitesimal vector of joint displacements.

You can write a dot product as a vector multiplication, so

$$\underbrace{\mathcal{F}^T \delta \mathcal{X}}_{\substack{\text{Cartesian} \\ \text{space}}} = \underbrace{\tau^T \delta \Theta}_{\substack{\text{Joint} \\ \text{space}}}$$

Recall that the Jacobian relates infinitesimal motions of the joints to infinitesimal motions of the end effector, e.g. $\delta \mathcal{X} = J \delta \Theta$. So we can write

$$\mathcal{F}^T J \delta \Theta = \tau^T \delta \Theta$$

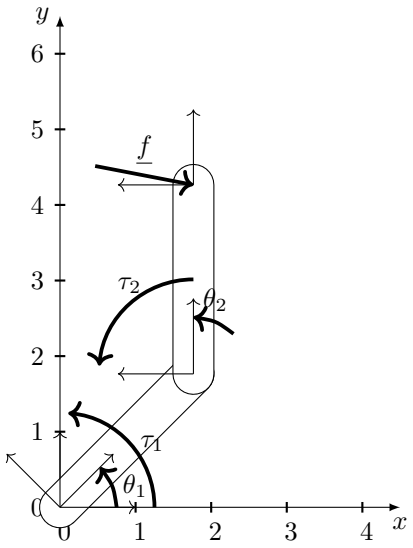
And this has to be true for **any** value of $\delta\Theta$, so we can cancel this term out to get

$$\mathcal{F}^T J = \tau^T$$

or

$$\tau = J^T \mathcal{F}$$

# 2 Example




Given $\underline{f}$, we want to find $\underline{\tau}$. From our previous result, we know that

$$\underline{\tau} = {}^0J^T \, {}^0\underline{f} = {}^3J^T \, {}^3\underline{f}$$

Normally, we get ${}^3\underline{f}$ from a force–torque sensor attached to the end effector, which is in frame 3. We get ${}^3J$ from Equation 5.66. So:

$$\underline{\tau} = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_1 \end{bmatrix}^T \begin{bmatrix} {}^3f_x \\ {}^3f_y \end{bmatrix}$$

or

$$\begin{aligned}\tau_1 &= l_1 s_2 f_x + (l_2 + l_1 c_2) f_y \\ \tau_2 &= l_2 f_y\end{aligned}$$

# 3 Cartesian transformation of velocities and forces

Sometimes we'll want to consider generalized force/moment and translational/rotational velocity vectors, e.g.

$$\mathcal{V} = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

and

$$\mathcal{F} = \begin{bmatrix} f \\ n \end{bmatrix}$$

We already have all the machinery we need to write $6 \times 6$ matrix transformations to handle these generalized cases. Specifically, we can rewrite Equations 5.45, 5.47 (for revolute joints):

$$\begin{aligned}{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \\ {}^{i+1}v_{i+1} &= {}^{i+1}_{i}R\left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}\right)\end{aligned}$$

or equation 5.48 (for prismatic joints):

$$\begin{aligned}{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i \\ {}^{i+1}v_{i+1} &= {}^{i+1}_{i}R\left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}\right) + \dot{d}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}\end{aligned}$$

IMPORTANT NOTE: we are dealing here with a static situation — we aren't considering motion of the robot's joints here — so we can simplify this as

$$\begin{aligned}{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i \\ {}^{i+1}v_{i+1} &= {}^{i+1}_{i}R\left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}\right)\end{aligned}$$

and

$$\begin{aligned}{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i \\ {}^{i+1}v_{i+1} &= {}^{i+1}_{i}R\left({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}\right)\end{aligned}$$

By factoring these results and integrating them into a single matrix–vector equation, we get (for revolute joints):

$$\begin{bmatrix} {}^{B}v_B \\ {}^{B}\omega_B \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}R & -{}^{B}_{A}R\,{}^{A}P_{B,Org}\times \\ 0 & {}^{B}_{A}R \end{bmatrix} \begin{bmatrix} {}^{A}v_A \\ {}^{A}\omega_A \end{bmatrix}$$

for two frames $\{A\} = i$ and $\{B\} = i + 1$, where we have expressed the cross product operator in matrix form:

$$P\times = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}$$

Note also that we have made use of the identity $\omega \times p = -p \times \omega$ in the $(1,2)$ element of this matrix expression.

We will refer to the matrix above as a **velocity transformation** $T_v$:

$$^{B}\mathcal{V}_B = {}^{B}_{A}T_v \, {}^{A}\mathcal{V}_A.$$

Be careful not to confuse this with a generalized coordinate transformation matrix, which is often denoted as ${}^{i+1}_{i}T$ !

In exercise 5.1 you are asked to derive the expression for $T_v^{-1}$, which is

$$\begin{bmatrix} {}^{A}v_A \\ {}^{A}\omega_A \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & P_{B,Org} \times {}^{A}_{B}R \, {}^{A} \\ 0 & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}v_B \\ {}^{B}\omega_B \end{bmatrix}$$

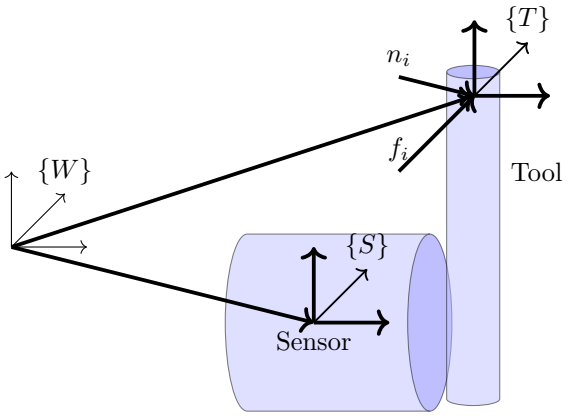
And similarly, from Equations 5.80 and 5.81, we can express the generalized force/moment transformation as

$$\begin{bmatrix} {}^{A}f_A \\ {}^{A}n_A \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}R & 0 \\ {}^{A}P_{B,Org} \times {}^{A}_{B}R & {}^{A}_{B}R \end{bmatrix} \begin{bmatrix} {}^{B}f_B \\ {}^{B}n_B \end{bmatrix}$$

We will refer to the matrix above as a **force–moment transformation** $T_f$:

$$^{A}\mathcal{F}_A = {}^{A}_{B}T_f \, {}^{B}\mathcal{F}_B.$$

## 3.1 Example




It's often the case that we will have a force–torque sensor mounted somewhere in the wrist of a robot, but that what we are really interested is the forces and torques applied at the tool tip, not those in the sensor frame. To transform from sensed forces and torques to the tool frame, we calculate ${}^{T}_{S}T_f$:

$${}^{T}_{S}T_f = \begin{bmatrix} {}^{T}_{S}R & 0 \\ {}^{T}P_{S,Org} \times {}^{T}_{S}R & {}^{T}_{S}R \end{bmatrix}$$

and therefore

$$^{T}\mathcal{F}_T = {}^{T}_{S}T_f \, {}^{S}\mathcal{F}_S$$