

Velocity Kinematics and Static Force Analysis

Up to this point we have considered only static positioning problems for a robotic manipulator. We now start to introduce the idea of motion into our analysis.

As a first step we consider the problem of deriving the **velocity** of the end effector for some arbitrary manipulator

To do this we will need a thorough understanding kinematic velocity analysis, and the tools involved: frames, moving frames, time derivatives of vectors, linear and angular velocities and their different representations.

What will turn out to be important is the relationship between the velocity in Cartesian space of the end effector and the velocity of the robot configuration in joint space.

The Jacobian, a matrix of derivatives, will be introduced as the key representation of this relationship

It will turn out that we can also achieve a static force/torque analysis of the manipulator using the Jacobian, and we will also explore this idea.

Structure of this Section

Thus the structure of the following lecture notes is:

- velocity of a single point
- velocity of a rigid body
- velocity analysis of a robotic manipulator
- the Jacobian
- Singularities
- Static Force Analysis

Velocity of a Point in Space

Consider the a vector Q expressed in frame $\{B\}$, ie BQ , that represents the position of a particle in space.

Then the velocity of of that particle, BV_Q , can be determined by taking the time derivative:

$${}^B V_Q = \frac{d}{dt} {}^B Q = \lim_{\delta t \rightarrow 0} \frac{{}^B Q(t + \delta t) - {}^B Q(t)}{\delta t}$$

Then clearly,

- The velocity of Q must be measured relative to some frame of reference - in this case frame $\{B\}$
- If the point is stationary in frame $\{B\}$ then the derivative of ${}^B Q$ in frame B is zero. This is the velocity of the point relative to $\{B\}$.
- If there exists some other frame $\{A\}$, then note that we can express the velocity of the point Q relative to $\{B\}$ in terms of frame $\{A\}$.

Notation

There are three important pieces of information to remember in the calculation of a linear velocity:

1. What is the point of which the velocity is measured.
2. Relative to what frame of reference is the velocity measured.
3. In what frame of reference is velocity expressed.

The notation in Craig attempts to code these three pieces of information

$${}^A ({}^B V_Q) = {}^A \frac{d}{dt} {}^B Q$$

Here, the linear velocity of the point Q is considered. The velocity is measured relative to the frame of reference B and is expressed in the frame of reference A .

$${}^A ({}^B V_Q) = {}^A_B R \frac{d}{dt} {}^B Q$$

where the right-hand-side may be explicitly calculated

$$\frac{d}{dt} {}^B Q = \left(\frac{d}{dt} ({}^B Q_X), \frac{d}{dt} ({}^B Q_Y), \frac{d}{dt} ({}^B Q_Z) \right)^T$$

The notation ${}^A ({}^B V_Q)$ is cumbersome and unwieldy. In most situations one of the superscripts is dropped.

Unfortunately, the superscript that is most natural to drop is the inner superscript, referring to which frame of reference the velocity is measured relative to.

Thus, one writes ${}^A V_Q$ where the fact that the velocity V_Q is measured relative to frame $\{B\}$

$${}^A V_Q = {}^A_B R \frac{d}{dt} {}^B Q$$

is understood from context.

Be careful of this notation as it is liable to cause considerable confusion.

Almost always, the velocity considered is the inertial velocity of the point considered. Thus, almost always you should read

$${}^A V_Q = {}^A_0 R \frac{d}{dt} {}^0 Q$$

where $\{0\}$ is the inertial frame.

Velocities of a frame of reference (of a rigid body)

Now let us determine the motion (velocity) of a rigid body. The first step is to attach a frame to the body.

Consider a point P on a rigid body whose position is given by ${}^B P$, where B is the rigid-body-attached frame.

Assume we have an inertial frame A . Then

$$\begin{aligned} {}^A V_P &= \frac{d}{dt} {}^A P \\ &= \frac{d}{dt} ({}^A P_{Borg} + {}^A_B R {}^B P) \\ &= \frac{d}{dt} {}^A P_{Borg} + \frac{d}{dt} {}^A_B R {}^B P \\ &= {}^A V_{Borg} + {}^A_B R \frac{d}{dt} {}^B P + \frac{d}{dt} ({}^A_B R) {}^B P \\ &= {}^A V_{Borg} + {}^A_B R {}^B V_P + ({}^A \Omega_B \times {}^A_B R {}^B P) \end{aligned}$$

where:

- ${}^A V_{Borg}$ is the velocity of origin of B with respect to A .
- ${}^A_B R {}^B V_P$ is the velocity of P wrt. B expressed in A .
- ${}^A \Omega_B \times {}^A_B R {}^B P$ is the velocity of P wrt. A due to the rotational velocity of B wrt. A .

Angular velocity.

We have introduced ${}^A\Omega_B$, but what is it?

${}^A\Omega_B$ is the angular velocity of a frame $\{B\}$ with respect to a frame of reference $\{A\}$. It can be represented by a triple of numbers:

$${}^A\Omega_B = [\Omega_1, \Omega_2, \Omega_3]^T$$

Its direction represents the instantaneous axis of rotation of frame $\{B\}$ with respect to a frame $\{A\}$, while its magnitude represents the speed of this rotation.

${}^A\Omega_B$ is obviously related to the derivative of the rotation matrix ${}^B_A R$, and we shall explore this relationship later.

An additional point is that, like any other vector, we can express ${}^A\Omega_B$ wrt. some other frame, say U , giving the cumbersome notation by Craig of

$${}^U({}^A\Omega_B)$$

where U is some absolute inertial frame, which leads to Craig introducing the notation

$$\omega_B = {}^U\Omega_B$$

when the identity of this absolute reference frame is clear.

We have also introduced the cross product in the final term of the equation on the previous slide for ${}^A V_P$, and we shall explore why we do this later.

Special Cases

First, let's check the special cases of rigid body motion:

- When B is purely translating with respect to A , ${}^A\Omega_B \times {}^A R^B P = 0$ since ${}^A\Omega_B = 0$. Hence

$${}^A V_P = {}^A V_{Borg} + {}^A R^B V_P$$

- When B is purely rotating with respect to A , then ${}^A V_{Borg} = 0$, and

$${}^A V_P = {}^A_B R^B V_P + ({}^A \Omega_B \times {}^A_B R^B P)$$

- When P is fixed with respect to B (usually the case if P is point on rigid body), then ${}^A_B R^B V_P = 0$, and

$${}^A V_P = {}^A V_{Borg} + ({}^A \Omega_B \times {}^A_B R^B P)$$

Summary

We can summarise the results so far as:

- The velocity of a point P is calculated as the time derivative of the position of P .
- This derivative must be taken with respect to some frame, say frame B , giving ${}^B V_P$
- The velocity vector can be express in terms of another frame, say A , so that we can have ${}^A({}^B V_P)$.
- When a point P is expressed in some frame B , and we wish to calculate its velocity (ie. take the derivative) with respect to some other frame A , then we get

$${}^A V_P = {}^A V_{Borg} + {}^A_B R^B V_P + ({}^A \Omega_B \times {}^A_B R^B P)$$

- This equation shows that ${}^A V_P$ derives from three things
 - the linear velocity of P wrt. the origin of B
 - the linear velocity of the origin of B wrt. the origin of A
 - the rotational velocity of B wrt. A
- the rotational velocity of B wrt. A is given by the angular velocity Ω , which represents both an axis and magnitude of rotation.

In the following slides we will focus in on the ${}^A \Omega_B \times {}^A_B R^B P$ term. We have introduced this term, but not really explained where it has come from. But first an example.

Example 5.1:

Skew symmetric matrices and the derivative of R

Rotation matrices are elements of the group of orthogonal matrices

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = I_3, \det(R) = 1\}$$

For the case where a frame, say B , is rotating with respect to another, say A , we are interested in a continuous dynamic change in a rotation matrix.

A continuous dynamic change in a rotation matrix is linked to the sequence of skew symmetric matrices. Why?

$$R^T R = I$$

$$\frac{d}{dt} R^T R = 0 = \dot{R}^T R + R^T \dot{R}$$

Thus,

$$R^T \dot{R} = - \left(R^T \dot{R} \right)^T = S$$

is a skew symmetric matrix (a skew symmetric matrix is one where elements $a_{ij} = -a_{ji}$ and $a_{ii} = 0$).

Let $R^T \dot{R} = S(t)$ be a time-varying skew symmetric matrix. Then

$$\dot{R} = RS(t)$$

This equation is termed the attitude *kinematics*. It relates a rotation R to its derivative through S .

Intuitively, S must “contain” the information on how R is changing over time.

S as the Angular Velocity Matrix

S is a skew symmetric matrix, and can be written in the form

$$S = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

where we call S the angular velocity matrix.

It can be shown (see Craig - Sect. 5.4) by direct differentiation of some rotation matrix ${}^A_B R$, that S can be written in the form

$$S = \begin{pmatrix} 0 & -k_z \dot{\theta} & k_y \dot{\theta} \\ k_z \dot{\theta} & 0 & -k_x \dot{\theta} \\ -k_y \dot{\theta} & k_x \dot{\theta} & 0 \end{pmatrix}$$

where $K = [k_x, k_y, k_z]^T$ is the unit vector passing through the origin of $\{A\}$ and $\{B\}$ with a line of action defining the axis of rotation and where $\dot{\theta}$ gives the magnitude of the rotational velocity.

So S does define the rotational velocity of frame $\{B\}$ wrt. $\{A\}$.

The Angular Velocity Vector Ω

We are interested in deriving the expression ${}^A\Omega_B \times {}^A_B R^B P$. Recall this was the component of ${}^A V$ due to the rotation of frame $\{B\}$ wrt. $\{A\}$.

If we take the derivative of

$$\begin{aligned}
 {}^A V_P &= \frac{d}{dt} {}^A P \\
 &= \frac{d}{dt} ({}^A P_{Borg} + {}^A_B R^B P) \\
 &= \frac{d}{dt} {}^A P_{Borg} + \frac{d}{dt} {}^A_B R^B P \\
 &= {}^A V_{Borg} + {}^A_B R \frac{d}{dt} {}^B P + \frac{d}{dt} ({}^A_B R) {}^B P \\
 &= {}^A V_{Borg} + {}^A_B R^B V_P + \frac{d}{dt} ({}^A_B R) {}^A_B R^{-1} {}^A P \\
 &= {}^A V_{Borg} + {}^A_B R^B V_P + S {}^A P \\
 &= {}^A V_{Borg} + {}^A_B R^B V_P + ({}^A\Omega_B \times {}^A P) \\
 &= {}^A V_{Borg} + {}^A_B R^B V_P + ({}^A\Omega_B \times {}^A_B R^B P)
 \end{aligned}$$

Where did Ω come from?

Where did the cross product come from?

A well known property of a skew symmetric matrix is that, if

$$S = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

and

$$\Omega = [\omega_1, \omega_2, \omega_3]^T$$

then, for some vector ${}^A P$,

$$S^A P = \Omega \times {}^A P$$

where we have seen previously that Ω , say ${}^A \Omega_B$, is the angular velocity vector which has a direction representing the instantaneous axis of rotation of frame $\{B\}$ with respect to a frame $\{A\}$, and a magnitude representing the speed of this rotation.

Wedge Product

Any skew symmetric matrix Ω (in $\mathbb{R}^{3 \times 3}$) may be written

$$\Omega = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}$$

Thus, Ω is parameterised by three components

$$\omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

Notation

$$\Omega = \omega^\wedge$$

Note that

$$\omega \times v = \begin{pmatrix} \omega_2 v_3 - v_2 \omega_3 \\ \omega_3 v_1 - \omega_1 v_3 \\ \omega_1 v_2 - \omega_2 v_1 \end{pmatrix} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} v = \omega^\wedge v$$

The skew parameterisation ω^\wedge is equivalent to the matrix representation of the linear operator associated with the vector cross product.

So you may see ${}^A V_P$ written alternatively as:

$${}^A V_P = {}^A V_{Borg} + {}^A_B R^B V_P + ({}^A \Omega_B \wedge {}^A_B R^B P)$$

Attitude Kinematics Revisited.

Recall the attitude kinematics

$$\frac{d}{dt} {}^A_B R = {}^A_B R S(t)$$

obtained by differentiating $R^T R = I_3$.

Note that one could also differentiate $RR^T = I_3$ to obtain a relation

$$\frac{d}{dt} {}^A_B R = \bar{S}(t) {}^A_B R$$

This leads to a relationship

$${}^A_B R S(t) {}^A_B R^T = \bar{S}(t)$$

- S relates the angular velocity of frame $\{B\}$ with respect to frame $\{A\}$ written with respect to frame $\{B\}$.
- Conversely, \bar{S} relates the angular velocity of frame $\{B\}$ with respect to frame $\{A\}$ written with respect to frame $\{A\}$.

Alternative representations to “Angle-axis” approach: Euler angles

The attitude kinematics may also be expressed in terms of the other representations.

Consider the Z-Y-X Euler angles

$${}^A_B R := R_\phi R_\theta R_\psi$$

The angular velocity of $\{B\}$ with respect to $\{A\}$, expressed in frame $\{A\}$ is the sum of the components due to the rotation of each individual Euler angle

$${}^A \Omega_B = \dot{\phi} e_3 + \dot{\theta} R_\phi e_2 + \dot{\psi} R_\phi R_\theta e_1.$$

- The first term is associated with the rotation ϕ taken around the $Z_A \in \mathcal{A}$ axis, where Z_A is just the e_3 co-ordinate vector.
- The θ rotation occurs around $R_\phi e_2$ the rotated version of Y_1 . The Y axis of the rotated frame after the yaw rotation has been applied.
- Finally the roll ψ occurs around the axis $R_\phi R_\theta e_1$, the image of X_A under the first two rotations

Computing these rotations yields

$$\begin{aligned} {}^A \Omega_B &= \dot{\phi} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \dot{\theta} \begin{pmatrix} -s_\phi \\ c_\phi \\ 0 \end{pmatrix} + \dot{\psi} \begin{pmatrix} c_\theta c_\phi \\ c_\theta s_\phi \\ s_\theta \end{pmatrix} \\ &= \begin{pmatrix} 0 & -s_\phi & c_\theta c_\phi \\ 0 & c_\phi & c_\theta s_\phi \\ 1 & 0 & -s_\theta \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = W(\phi, \theta, \psi) \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \end{aligned}$$

From the above we have

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = W^{-1}(\phi, \theta, \psi)^A \Omega_B.$$

This equation is termed the **attitude kinematics for the Z-Y-X Euler angles**. It denotes the rate of change of the Euler angles given a known angular velocity.

The inverse of ${}^B W$ is given by

$${}^B W^{-1} := \frac{1}{\cos(\theta)} \begin{pmatrix} 0 & s_\psi & c_\psi \\ 0 & c_\theta c_\psi & -c_\theta s_\psi \\ c_\theta & s_\theta s_\psi & s_\theta c_\psi \end{pmatrix}.$$

Note that

$$\det(W) = -\cos(\theta)$$

and thus the Euler angle attitude kinematics are not valid where $\theta = \frac{\pi}{2}$. This singularity is related to the non-uniqueness of the Euler angle representation.

In the frame $\{B\}$ the angular velocity is given by

$${}^B \omega = {}^B_A R W(\phi, \theta, \psi) \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = {}^B W(\phi, \theta, \psi) \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}$$

where

$${}^B W(\phi, \theta, \psi) = \begin{pmatrix} -s_\theta & 0 & 1 \\ c_\theta s_\psi & c_\psi & 0 \\ c_\theta c_\psi & -s_\psi & 0 \end{pmatrix}$$

Velocity analysis of a manipulator

Now apply the velocity kinematic tools seen thus far to the specific case of a robotics manipulator.

In computing the velocity kinematics of manipulator the aim is to express the velocity of the end effector with respect to the station or base frame, which is a non-moving, or *inertial* frame.

The joint velocities $\dot{\theta}_i$ or \dot{d}_i will provide information about the velocity of the frame i with respect to frame $i - 1$.

In general, the velocity, $d/dt^0 P$ comprises a component due to the inertial velocity of the base frame $\{A\}$ (if it is non-zero), a component due to the velocity of $\{B\}$ relative to $\{A\}$ (rotational and linear) and the velocity of ${}^B P$.

In each case the actual velocity vector V of some point on the end effector is a vector that can be expressed in any of the frames of reference

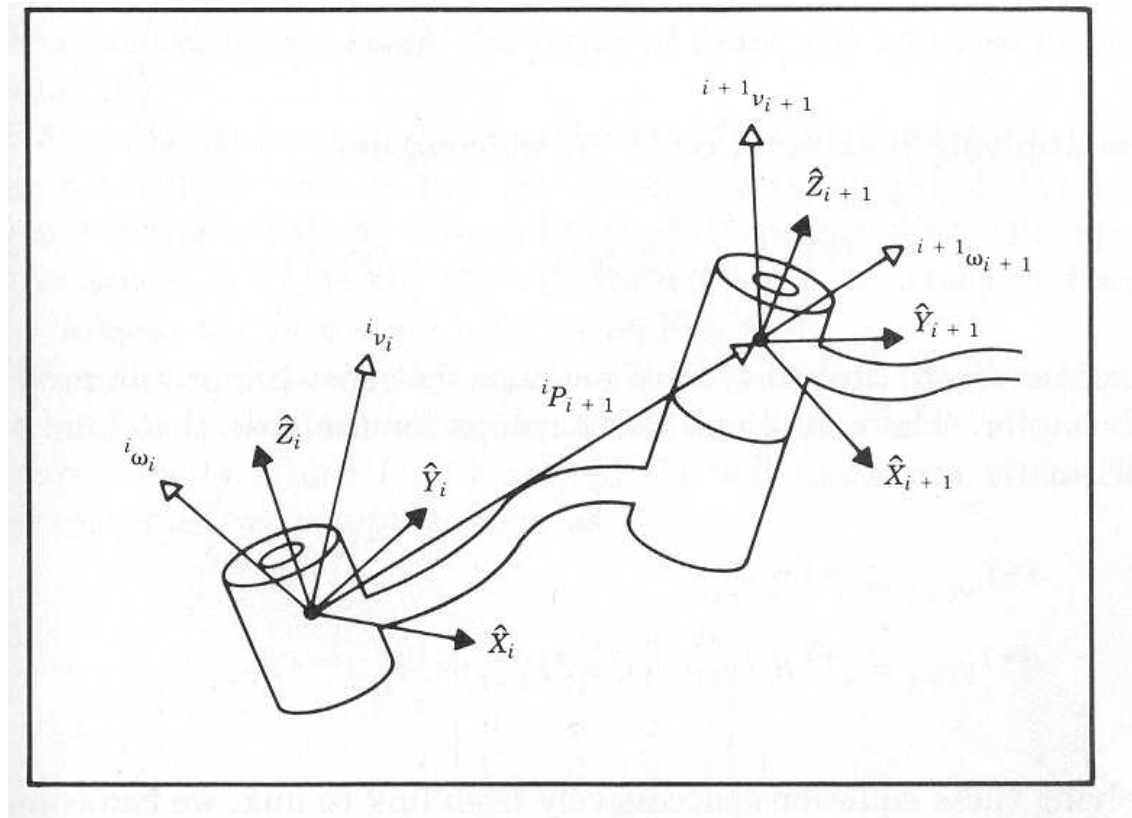
$${}^A V_P = {}^A_B R {}^B V_P, \quad {}^0 V_P = {}^0_A R {}^A_B R {}^B V_P$$

Note : in order to extend a kinematic model to a dynamic model, velocities must be measured relative to some inertial frame of reference.

Although the velocity is the inertial velocity - it is often written in the local frame of reference.

Motion of Robot links

Consider the motion of link i relative to link $i - 1$.



Remember that linear velocity is associated with a point and angular velocity with a body (frame); hence the linear velocity of a link means the linear velocity of the origin of the link frame and the angular velocity of a link means the angular velocity of the link frame.

What is the relationship between ${}^i V_i, {}^i \omega_i$ and ${}^{i+1} V_{i+1}, {}^{i+1} \omega_{i+1}$?

Lets look at the relative linear and angular velocities that occur between the manipulator's links.

Relative Linear Velocity between Links

Consider the motion of link i relative to link $i - 1$.

The **linear velocity** of i is

$$\begin{aligned}
 {}^{i-1}V_i &= \frac{d}{dt} {}^{i-1}P_i = \frac{d}{dt} {}^{i-1}T(d_i, \theta_i) {}^iP_i \\
 &= \frac{d}{dt} \begin{pmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \\
 &= \frac{d}{dt} \begin{pmatrix} a_{i-1} \\ s\alpha_{i-1}d_i \\ c\alpha_{i-1}d_i \\ 1 \end{pmatrix} = \dot{d}_i \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \\
 &= \dot{d}_i Z_i
 \end{aligned}$$

where iP_i is the position of the origin of frame i in frame i , and ${}^{i-1}T(d_i, \theta_i)$ is the link $i - 1$ to i homogeneous transformation.

In retrospect this is no surprise since the point ${}^{i-1}P_i$ is fixed for revolute joints and can only move in the joint axis direction Z_i for prismatic joints.

Relative Angular Velocity between Links

Using the geometric insights of above, the angular velocity of link i relative to link $i - 1$ is given by

$${}^{i-1}\omega_i = \dot{\theta}_i {}^{i-1}Z_i$$

So, to summarise, for the two types of joint, either

$${}^{i-1}V_i = \dot{d}_i {}^{i-1}Z_i, \quad \text{or} \quad {}^{i-1}\omega_i = \dot{\theta}_i {}^{i-1}Z_i.$$

The above expressions both relate the kinematic velocity of link i relative to link $i - 1$. In practice, one wishes to understand the inertial velocity of link i measured relative to the station frame.

Thus, the velocity of $\{i\}$ should be calculated relative to $\{0\}$.

Inertial linear velocity of link i

The inertial velocity of link i measured relative to an inertial frame can be derived from the relative link velocity $({}^{i-1}V_i, {}^{i-1}\omega_i)$ and the inertial velocity $({}^0V_{i-1})$ of link $i - 1$.

Note that velocity components, either linear or angular, sum as long as they are expressed in the same frame of reference.

A geometric argument gives

$${}^{i-1}({}^0V_i) = ({}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) + {}^{i-1}({}^0V_{i-1}) + \dot{d}_i {}^{i-1}Z_i$$

- The term $({}^{i-1}\omega_{i-1} \times {}^{i-1}P_i)$ is due to the point ${}^{i-1}P_i$ rotating around the origin of the frame $\{i - 1\}$ with its angular velocity. Note that ${}^{i-1}\omega_{i-1}$ is the angular velocity of frame $\{i - 1\}$ and need not lie in direction ${}^{i-1}Z_{i-1}$ since it is derived from all the previous rotation movements of the robotic manipulator.
- The term ${}^{i-1}({}^0V_{i-1})$ is the linear velocity of frame $\{i - 1\}$.
- The term $\dot{d}_i {}^{i-1}Z_i$ is the linear velocity of frame $\{i\}$ expressed in frame $\{i - 1\}$.
- One of either the first or third terms in this equation will be zero depending on whether the joint in question is prismatic or revolute.

To save notation we will write ${}^p v_q = {}^p({}^0V_q)$ to denote the inertial velocity of link q expressed in frame p . Thus,

$${}^{i-1}v_i = ({}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) + {}^{i-1}v_{i-1} + \dot{d}_i {}^{i-1}Z_i$$

Inertial angular velocity of link i

Let

$${}^p\omega_q = {}^p({}^0\omega_q)$$

denote the inertial angular velocity of link q expressed in frame $\{p\}$.

A geometric argument gives

$${}^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + \dot{\theta}_i {}^{i-1}Z_i$$

This is a direct consequence of summation of angular velocities in frame $i - 1$.

Iterative solution for velocity of link i

We have seen that the inertial velocities of a link i can be expressed in terms of the inertial velocities of link $i - 1$ and the relative velocities of link i wrt. link $i - 1$.

We are interested in determining the inertial velocity of the end effector, expressed in some frame, usually the frame attached to the base of the robot (which is usually inertial).

This suggests an iterative scheme where we iteratively propagate link velocities, starting from the robot base, moving one link at a time until we reach the end effector link.

The velocity of link i relative to the inertial base frame $\{0\}$ can be calculated iteratively as follows:

1. Set ${}^0\omega_0 = 0$ and ${}^0v_0 = 0$. Thus, the initial reference frame is stationary. Set $i = 0$.
2. Increment i by one. (The index i takes the values $i = 1, \dots, n$). Compute the link i velocity in frame $\{i-1\}$ according to the iterative equations,

$$\begin{aligned} {}^{i-1}\omega_i &= {}^{i-1}\omega_{i-1} + \dot{\theta}_i {}^{i-1}Z_i \\ {}^{i-1}v_i &= {}^{i-1}v_{i-1} + ({}^{i-1}\omega_{i-1} \times {}^{i-1}P_i) + \dot{d}_i {}^{i-1}Z_i. \end{aligned}$$

3. If $i = n$ then set

$${}^0\omega_n = {}^0_{n-1}R^{n-1}\omega_n, \quad {}^0v_n = {}^0_{n-1}R^{n-1}v_n.$$

Otherwise, transform the computed link velocity from frame $\{i-1\}$ to frame $\{i\}$

$$\begin{aligned} {}^i\omega_i &= {}^i_{i-1}R {}^{i-1}\omega_i \\ {}^i v_i &= {}^i_{i-1}R {}^{i-1}v_i \end{aligned}$$

and return to step 2).

Example 5.2:

Jacobians

In general a Jacobian is simply a matrix representation of a multi-dimensional derivative.

Let f_1, f_2, \dots, f_m each be a some differentiable functions mapping $\mathbb{R}^1 \rightarrow \mathbb{R}^m$

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n) \\ y_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ y_m &= f_m(x_1, x_2, \dots, x_n) \end{aligned}$$

Then we can write all the derivatives, ie. of each y_i with respect to each x_j , as a matrix

$$J_f(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(x)}{\partial x_1} & \dots & \frac{\partial f_m(x)}{\partial x_n} \end{pmatrix}$$

where $x = x_1, x_2, \dots, x_n$, and where $J_f(x)$ is called the Jacobian.

Velocity Jacobian of a Manipulator

Up to this point we have determined methods to derive the linear v and angular ω velocity of the robot end effector.

We have found that we derive expressions for v and w that include the joint angles/extensions q_1, q_2, \dots, q_n and joint rates $\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n$.

It turns out that we can write down these expressions in a general way as

$$v = J_p(q_1, \dots, q_n)\dot{q}$$

and

$$\omega = J_o(q_1, \dots, q_n)\dot{q}$$

where, for a 6-dof manipulator ($n = 6$):

- v is the (3x1) vector of linear velocities in the x, y and z directions.
- w is the (3x1) vector of angular velocities around the x, y and z directions.
- $\dot{q} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_6]^T$, the vector of joint velocities.
- J_p is a jacobian; a (3x6) matrix governing the contribution of joint velocities \dot{q} to the end effector **linear** velocity.
- J_o is also a jacobian; a (3x6) matrix governing the contribution of joint velocities \dot{q} to the end effector **angular** velocity.

The usual approach is to stack these Jacobians on top of one another to form the equation for the **Cartesian** velocity \mathcal{V} as:

$$\mathcal{V} = \begin{pmatrix} v \\ \omega \end{pmatrix} = J(q_1, \dots, q_n)\dot{q}$$

where we denote

$$J = \begin{pmatrix} J_p \\ J_o \end{pmatrix}$$

as the **velocity Jacobian**

The key point to remember is that the velocity Jacobian is the link between the Cartesian velocity of the wrist frame and time rate of change of the joint variables (joint velocities).

The force Jacobian is linked to the velocity Jacobian by the principal of conservation of energy.

Things to remember about the Jacobian:

It is a linear mapping, ie. J is a linear function of q_1, \dots, q_n

J will change at each new set of q_1, \dots, q_n . From a practical point of view this means we need to recompute (not rederive!) the Jacobian at each new robot configuration.

Computing the Velocity Jacobian of a manipulator

Let the Jacobian J be partitioned into (3×1) column vectors, ie.

$$J = \begin{pmatrix} J_{p_1} \dots J_{p_i} \dots J_{p_n} \\ J_{o_1} \dots J_{o_i} \dots J_{o_n} \end{pmatrix}$$

and we state again that

$$\dot{q} = [\dot{q}_1, \dots, \dot{q}_i, \dots, \dot{q}_n]^T$$

is the joint velocity vector.

Then the term $\dot{q}_i J_{p_i}$ represents the contribution of single joint i to the end-effector linear velocity, while the term $\dot{q}_i J_{o_i}$ represents the contribution of single joint i to the end-effector angular velocity.

Then using this fact, we can write down each J_{p_i} and J_{o_i} depending on whether joint i is prismatic or revolute.

More specifically, if joint i is prismatic

$$\begin{pmatrix} J_{p_i} \\ J_{o_i} \end{pmatrix} = \begin{pmatrix} {}^0Z_i \\ 0 \end{pmatrix}$$

or if joint i is revolute then

$$\begin{pmatrix} J_{p_i} \\ J_{o_i} \end{pmatrix} = \begin{pmatrix} {}^0Z_i \times ({}^0p - {}^0p_i) \\ {}^0Z_i \end{pmatrix}$$

0Z_i is the unit vector along the joint axis of joint i expressed in the inertial frame.

0p is the position of the point we are interested in determining the velocity for (the end effector) and 0p_i is the position of the origin of the i^{th} frame.

Note that given we have the forward kinematic transformations of a manipulator, ie. ${}^0T_1, {}^0T_2, \dots, {}^0T_N$, we can read off the values for ${}^0p_1, {}^0p_2, \dots, {}^0p$ as the first three elements of the fourth columns of these matrices respectively.

The above determination of the Jacobian was on the basis of geometric intuition.

It is also interesting to note that one can derive the Jacobian directly by differentiating the forward kinematic equations with respect to the joint variables - but we will not do it here :-)

Frames of Reference and the Velocity Jacobians

The velocity Jacobian $J(q)$ must be identified with a frame of reference in which the Cartesian velocity \mathcal{V} finally expressed.

Up to this point we have assumed that the Jacobian is identified with the base frame $\{0\}$, ie ${}^0J(q)$.

It is possible that we may want to convert the Jacobian to be relevant to another frame, so that our end effector Cartesian velocity calculations are output with respect to another frame (say the station frame).

First note that a frame transformation on a (6x1) Cartesian velocity vector may be achieved according to

$$\begin{pmatrix} {}^Av \\ {}^A\omega \end{pmatrix} = \begin{pmatrix} {}^A_B R & 0 \\ 0 & {}^A_B R \end{pmatrix} \begin{pmatrix} {}^Bv \\ {}^B\omega \end{pmatrix}$$

so that

$$\begin{pmatrix} {}^Av \\ {}^A\omega \end{pmatrix} = \begin{pmatrix} {}^A_B R & 0 \\ 0 & {}^A_B R \end{pmatrix} {}^BJ(q)\dot{q}$$

and so

$${}^A J(q) = \begin{pmatrix} {}^A_B R & 0 \\ 0 & {}^A_B R \end{pmatrix} {}^B J(q)$$

Example 5.3:

The Jacobian and trajectory tracking

Consider a smooth trajectory a of a robot expressed in Cartesian coordinates of the end effector

$$a : t \mapsto {}^0_n T(t).$$

The velocity of the end effector is

$$\frac{d}{dt} {}^0_n T(t) = {}^0_n \mathcal{V}(t)$$

From the velocity Jacobian one has

$${}^0_n \mathcal{V}(t) = \begin{pmatrix} {}^0 v_n(t) \\ {}^0 \omega_n(t) \end{pmatrix} = J(q) \dot{q}$$

Thus, to track the desired trajectory it is sufficient to assign a speed controller to joints to achieve

$$\dot{q} = J^{-1}(q) \begin{pmatrix} {}^0 \omega_n(t) \\ {}^0 v_n(t) \end{pmatrix}$$

Hence, in robotics, we are most often interested in the inverse of the Jacobian.

Inverting the velocity Jacobian.

In a typical industrial manipulator there are exactly the same number of active joint variables as degrees of freedom.

For a full 6DOF manipulator moving in $SE(3)$ then there are 6 active joint variables $q = (q_1, \dots, q_6)$.

As a consequence

$$J(q) \in \mathbb{R}^{6 \times 6}$$

is a square matrix and

$$\begin{aligned}\dot{q} &= J^{-1}(q) \begin{pmatrix} {}^0\omega_n(t) \\ {}^0v_n(t) \end{pmatrix} \\ &= J^{-1}(q) {}^0\mathcal{V}_n(t)\end{aligned}$$

is defined as long as $J(q)$ is invertible.

Dealing with under-actuated manipulators.

If there are insufficient joints actuated, say $p < 6$ joints, to achieve full 6DOF movement the situation becomes somewhat more serious.

In this case

$$J(q) \in \mathbb{R}^{p \times 6}$$

is rank p .

To solve the tracking problem one requires that ${}^0\mathcal{V}_n(t)$ lies in the range space of $J(q)$ in order that

$$J(q)\dot{q} = {}^0\mathcal{V}_n(t)$$

has a solution \dot{q} .

The range space of $J(q) = \text{span}\{\text{col}J(q)\}$ is the subspace of accessible velocities for the manipulator. It is associated with tangent direction to the manipulator subspace for an under-actuated manipulator.

If the velocity profile desired is derived from differentiating a smooth curve in the forward kinematics (one that satisfies the manipulator subspace constraints) then the velocity constraint is automatically satisfied.

Singularities

Even in the case where there are 6 joints for 6DOF robotic manipulator there can be problems in solving

$$\dot{q} = J^{-1}(q) {}^0\mathcal{V}_n(t)$$

This equation cannot be solved if $J(q)$ becomes singular. That is, if the range space of $J(q)$ degenerates in at least one direction. Denote the degenerate direction of $J(q)$ at a singular point by \bar{v} , then

$$\bar{v}^T J(q)\dot{q} = 0 = \bar{v}^T {}^0\mathcal{V}_n(t)$$

for all \dot{q} . Thus, ${}^0\mathcal{V}_n(t)$ must be orthogonal to \bar{v} . Such points are called **singularities of the mechanism**.

Singularities are a fundamental part of an actuator mechanism.

1. They represent configurations at which the mobility of the mechanism is reduced, ie. it is not possible to impose an arbitrary motion on the end effector.
2. In the neighbourhood of a singularity, small velocities in the operational space may cause large velocities in the joint space.

Types of Singularities

1. **Workspace boundary singularities:** It is clear that the end link of a manipulator cannot be moved beyond the boundary of its workspace. All boundary points must be singular points of the manipulator. The singular direction \bar{v} is the normal to the workspace boundary.
2. **Workspace interior singularities:** Singularities are also possible within the workspace. They are usually due to the alignment of two or more joint axes eg. when joint axis 5 on Puma is zero.

Example 5.4:

Locating the singularities of a manipulator

To locate all the singularities of a manipulator

1. Use your physical insight of the manipulator geometry and workspace to identify workspace boundary singularities and interior point singularities.
2. Compute the determinant of the velocity Jacobian and find q such that

$$\det(J(q)) = 0$$

This equation leads to complicated non-linear equations in q that often cannot be solved in closed-form. This approach can be used for simple manipulators but is most often used as confirmation of singularity rather than a means to find singular points.

Trajectory tracking in the vicinity of singularities

Consider a 6DOF manipulator moving within its workspace in $SE(3)$. Let ${}^0\mathcal{V}_n(t)$ be a bounded velocity profile of a trajectory that passes close to a singularity but does not pass through the singularity.

Thus,

$$|{}^0\mathcal{V}_n(t)| < B_0$$

for some bound $B_0 > 0$ and

$$\det(J(q)) > \epsilon$$

for $\epsilon > 0$. Note that ϵ may be small.

The joint velocities for trajectory tracking are well defined at all points on the trajectory.

$$\dot{q} = J^{-1}(q)^0 \mathcal{V}_n(t)$$

Due to the inverse in the Jacobian the best general bound on the joint velocities that is possible is

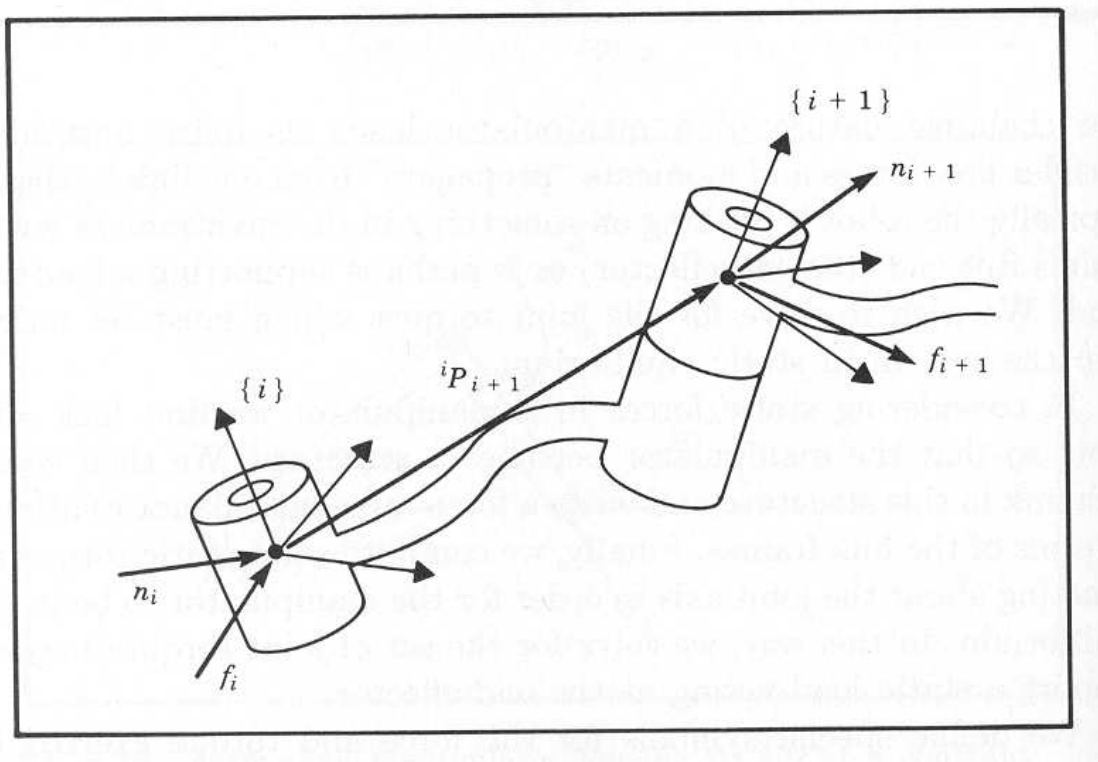
$$|\dot{q}| \leq \frac{B_0}{\epsilon}$$

The potentially large values of \dot{q} generated by the trajectory tracking algorithm may easily violate the joint motor capabilities. This is a natural consequence of operating close the structural limitations of a robotic manipulator and one must be careful.

Static Force/Torque Analysis of Manipulators.

Imagine a non-moving manipulator holding a load in its end effector, or pushing against a wall, etc.

We can conduct a “statics” force/torque analysis - ie lock all the joints and calculate the force and torque that each link must exert on the next for the structure to remain in its locked position.



This statics analysis can be done iteratively according to:

Let

f_i = force exerted on link i by link $i - 1$

n_i = torque exerted on link i by link $i - 1$

The static contact force balance equation (ignoring gravitational effects) for link i is

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

The static torque balance equation must be computed around a point. The point chosen is the origin of frame i . Thus, there is a component due to the transmitted torque plus a component due to the linear force ${}^i f_{i+1}$ applied at ${}^i P_{i+1}$

$${}^i n_i - {}^i n_{i+1} - ({}^i P_{i+1} \times {}^i f_{i+1}) = 0.$$

This leads to a coupled pair of iterative equations relating forces and moments applied to link $\{i + 1\}$ to forces and moments applied to link $\{i\}$.

Iterative algorithms for static contact forces.

Define the generalised force vector as $\mathcal{F} = (F, N)$, where F is the linear force and N is the rotational torque applied by the origin contact point P_{T_0} of the tool frame.

The generalised force applied at each active joint variable due to a force $\mathcal{F} = (F, N)$ applied at link i can be calculated iteratively:

1. Set ${}^n f_n = F$ and ${}^n n_n = N$. Thus, the initial reference frame is taken as the end effector frame (or tool frame). This force $\mathcal{F} = (F, N)$ must be specified in frame $\{n\}$.
2. Decrement i by one. (The index i takes the values $i = n - 1, \dots, 0$). Compute the link i force in frame $\{i\}$ according to the iterative equations,

$${}^i f_i = {}^i f_{i+1} = {}^i_{i+1} R ({}^{i+1} f_{i+1})$$

$${}^i n_i = {}^i n_{i+1} + ({}^i P_{i+1} \times {}^i f_{i+1}) = {}^i_{i+1} R ({}^{i+1} n_{i+1}) + ({}^i P_{i+1} \times {}^i f_{i+1})$$

3. **If** the joint is revolute, set

$$\tau_i = {}^i n_i^T {}^i Z_i = \langle {}^i e_3, {}^i n_i \rangle.$$

Else the joint is prismatic. Set

$$\tau_i = {}^i f_i^T {}^i Z_i = \langle {}^i e_3, {}^i f_i \rangle.$$

4. **If** $i > 0$ return to step 2). **Otherwise** end.

Example 5.5:

The Force Jacobian

The **force Jacobian** of a manipulator is defined to be the algebraic mapping between generalised joint forces (torques or linear forces for revolute or prismatic joint respectively) and the Euclidean forces applied at the end effector of the robot.

Let $\mathcal{F} = (F, N)$ be the linear force (F) and rotational torque N applied by the origin contact point P_{T_0} of the tool frame. Let $\tau = (\tau_1, \dots, \tau_n)$ denote the generalised forces at each active joint variable (q_1, \dots, q_n) .

$$\tau = \mathbb{J}\mathcal{F}$$

Of course \mathcal{F} must be expressed with respect to a frame of reference and \mathbb{J} must be expressed in the same frame of reference.

1. The force Jacobian can be computed via an iterative method similar to that used to compute the velocity Jacobian.
2. The velocity Jacobian of manipulator is also closely linked to the force Jacobian (we explore this further below).

The static force Jacobian concerns only the components of force and torque due to *contact of the manipulator with the external world*. Gravitational effects are ignored in the following discussion and must be pre-computed and compensated for in any control algorithm based on these ideas.

Principal of virtual work and duality of force and velocity.

The principal of virtual work ensures that the ‘infinitesimal’ work done by the end effector in the tool frame is equal to the total ‘infinitesimal’ work done by the sum of all the active joint variables.

The **principal of virtual work** yields

$$\langle \mathcal{F}, \delta X \rangle = \langle \tau, \delta q \rangle$$

where δX denotes an infinitesimal change in the Euclidean position of the end effector of the manipulator and δq denotes an infinitesimal change in the active joint variables.

Dividing through by δt and limiting one has

$$\frac{\delta X}{\delta t} = {}^n\mathcal{V}_n, \quad \frac{\delta q}{\delta t} = \dot{q}$$

Thus, the work balance equation from above may be written

$${}^0\mathcal{F}^T {}^0\mathcal{V}_n = \tau^T \dot{q}$$

where both the force exerted by the end effector and its Euclidean velocity must be expressed in the same frame of reference. The base frame of reference, $\{0\}$, is chosen for the convenience of the subsequent derivation.

Note: The joint coordinates have only the one representation as a vector of variables. No frame of reference need be specified.

Velocity Jacobian and the force Jacobian.

Within the workspace of the robot manipulator one has

$${}^0\mathcal{V}_n = J(q)\dot{q}$$

from the definition of the velocity Jacobian.

Substituting into the work balance equation one has

$${}^0\mathcal{F}^T J(q)\dot{q} = \tau^T \dot{q}$$

and since this is true for all \dot{q} then

$$J^T(q){}^0\mathcal{F} = \tau$$

Thus, the force Jacobian

$$\tau = \mathbb{J}\mathcal{F}$$

is given by $\mathbb{J} = J^T(q)$

$$\tau = J^T(q)\mathcal{F}.$$

This is an algebraic expression of the principal of duality of forces and velocities.

Note: The entire discussion undertaken above concerns only contact forces. Forces due to conservative gravitational effects can be explicitly incorporated into the iterative expression or modelled as a change in potential energy in the work balance equation.

Instantaneous transformation of velocities between frames of reference

We have been dealing with the Cartesian velocity \mathcal{V} of a rigid body, and we noted it is specified in some particular frame.

If we are interested in expressing \mathcal{V} in some other frame, we need a 6x6 transformation to achieve this mapping.

Consider a rigid-body with two different body-fixed frames $\{A\}$ and $\{B\}$ attached. The velocity of frame $\{B\}$ relative to frame $\{A\}$ is zero since the two frames are physically attached to the same rigid body.

The velocity of each frame of reference relative to a base frame $\{0\}$ may be non-zero. Let

$${}^X v_Y = {}^X ({}^0 V)_Y, \quad {}^X \omega_Y = {}^X ({}^0 \omega)_Y$$

for X, Y equal to either A or B

Recalling the velocity propagation equations from earlier one has

$$\begin{pmatrix} {}^B v_B \\ {}^B \omega_B \end{pmatrix} = \begin{pmatrix} {}^B_A R & -{}^B_A R {}^A P_{Borg}^\wedge \\ 0 & {}^B_A R \end{pmatrix} \begin{pmatrix} {}^A v_A \\ {}^A \omega_A \end{pmatrix}$$

where recall that ${}^A P_{Borg}^\wedge$ will be a skew symmetric matrix - the matrix operator corresponding to the cross product.

One writes

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

Clearly ${}^B T_v$ is a function of ${}^B T$ hence this result is only valid if there is no relative motion between frames A and B .

Alternatively, this result can be viewed as the **instantaneous** velocity transformation ${}^B T_v$ between two frames of reference if there is relative motion between them (ie. it is valid for a particular ${}^B T$).

Transformation of static forces between frames of reference

Using the principal of virtual work once more one has

$$\langle {}^A F, {}^A \mathcal{V}_A \rangle = \langle {}^B F, {}^B \mathcal{V}_B \rangle$$

since we assume that the two frames $\{A\}$ and $\{B\}$ are rigidly fixed together.

Thus, using the instantaneous velocity transformation ${}^B T_v$, one obtains

$${}^A F^T {}^A \mathcal{V}_A = {}^B F^T {}^B \mathcal{V}_B = {}^B F^T {}^B T_v {}^A \mathcal{V}_A$$

It follows that

$$\begin{aligned}
\begin{pmatrix} {}^A F_A \\ {}^A N_A \end{pmatrix} &= \begin{pmatrix} {}^B R_A & -{}^B R_A P_{B_0}^\wedge \\ 0 & {}^B R_A \end{pmatrix}^T \begin{pmatrix} {}^B F_B \\ {}^B N_B \end{pmatrix} \\
&= \begin{pmatrix} {}^B R_A^T & 0 \\ -({}^A P_{B_0}^\wedge)^T {}^B R_A^T & {}^B R_A^T \end{pmatrix} \begin{pmatrix} {}^B F_B \\ {}^B N_B \end{pmatrix} \\
&= \begin{pmatrix} {}^A R_B & 0 \\ {}^A P_{B_0}^\wedge {}^A R_B & {}^A R_B \end{pmatrix} \begin{pmatrix} {}^B F_B \\ {}^B N_B \end{pmatrix} \\
&= {}^A T_f^B \mathcal{F}_B
\end{aligned}$$

where ${}^A T_f^B$ is the **force-moment transformation**.

Summary

Initially looked at velocity kinematics of general rigid bodies with the purpose of applying the tools in this area to robotic manipulators eg. the 3 components of the velocity of a point in a moving frame.

Looked at deriving the velocity of the end effector of a robotic manipulator as a function of joint rates. Presented an iterative algorithm to do this velocity calculation - starting at the base and moving up to the end effector

Introduced the velocity Jacobian as the “mapping” between the Cartesian end effector velocity and joint velocities. Presented an algorithm to derive a velocity Jacobian.

Introduced the idea of a “static” force-torque analysis of a manipulator - in order to determine joint torques to be applied by the motors. Introduced an algorithm for determining the force/torques on each link - starting at the end effector and finishing at the base

Introduced the force Jacobian and investigated its relationship with the velocity Jacobian.