Aalto University
School of Electrical Engineering

Velocity relationships, part 1

ELEC-C1320 Robotics

Pekka Forsman

Topics

- Manipulator Jacobian matrix
  - Method 1: calculating the manipulator Jacobian matrix by differentiating the forward/direct kinematic equations of the mechanism
  - Method 2: calculating the manipulator Jacobian matrix by velocity propagation from link to link (geometric approach)
- Transforming velocities between coordinate frames

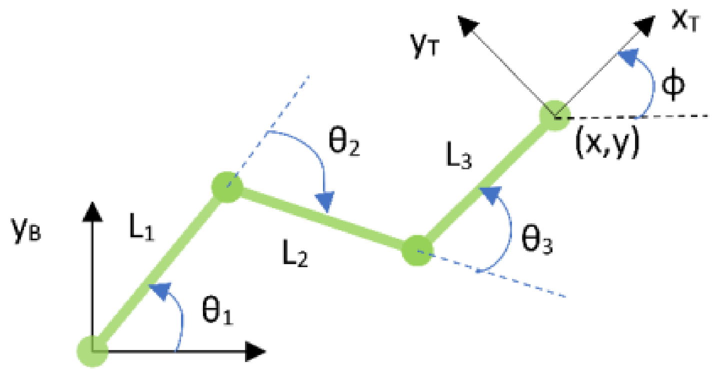
The theoretical field in robotics that discusses relations between motion (velocity) in manipulator joint space and motion (linear/angular velocity) in manipulator end-effector/tool (Cartesian) task space is called **Differential kinematics**.

The 3-dimensional end-effector pose $\xi \in SE(3)$ has a velocity which is represented by a 6-vector known as a spatial velocity. It has three vector elements for the linear velocity along the tool frame axes and three rotational velocity components around them, $\boldsymbol{v} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$. The joint space velocity and the end-effector velocity are related by the **manipulator Jacobian matrix** which is a **function of manipulator pose**.

Instantaneous velocity mappings between the joint and task spaces can be obtained through time derivation of the direct kinematics (numerical approach) or by velocity propagation from link to link (geometric approach).

Manipulator Jacobian matrix

We have a vector describing the robot tool pose in the task space (e.g. $\boldsymbol{y} = [y_1, y_2, y_3] = [x, y, \phi]$). Each of these coordinates is a function of robot joint configuration vector (e.g. $\boldsymbol{x} = [x_1, x_2, x_3] = [q_1, q_2, q_3] = [\theta_1, \theta_2, \theta_3]$). (*compare the text box to the right and the mechanism below*)



A Jacobian is the matrix equivalent of the derivative – the derivative of a vector-valued function of a vector with respect to a vector. If $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ then the Jacobian is the $m \times n$ matrix

$$J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}$$

The Jacobian is named after Carl Jacobi, and more details are given in Appendix E.

First we compute the derivative of $\boldsymbol{p}$ with respect to the joints variables $\boldsymbol{q}$. Since $\boldsymbol{p}$ and $\boldsymbol{q}$ are both vectors the derivative (*Corke, pp 230-231*)

$$\boxed{\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})} \qquad (8.1)$$

will be a matrix – a Jacobian matrix which is typically denoted by the symbol $\boldsymbol{J}$

To determine the relationship between joint velocity and end-effector velocity we rearrange Eq. 8.1 as

$$\boxed{\mathrm{d}\boldsymbol{p} = \boldsymbol{J}(\boldsymbol{q})\mathrm{d}\boldsymbol{q}}$$

and divide through by dt to obtain

$$\boxed{\begin{gathered}\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \boldsymbol{J}(\boldsymbol{q})\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \\ \dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}\end{gathered}}$$

The manipulator Jacobian matrix maps velocities from the joint coordinate space, or in other words, joint configuration space to the end-effector's Cartesian coordinate space and is itself a function of the joint coordinates.

If we consider the generic form equation of forward kinematics given in ch. 7

$$^{0}\xi = \mathcal{K}(\boldsymbol{q}) \qquad (7.4)$$

And differentiate both sides w.r.t time we get

$$^{0}\nu = {}^{0}\boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad (8.2)$$

where, for a six axes robot, the Cartesian velocity and joint velocity vectors would be (supercript 0 denotes to the reference frame of the Cartesian space velocity)

$$^{0}\boldsymbol{v} = [v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T$$

$$^{0}\dot{\boldsymbol{q}} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3 \quad \dot{q}_4 \quad \dot{q}_5 \quad \dot{q}_6]^T$$

There are different methods for forming the manipulator Jacobiaan matrix.

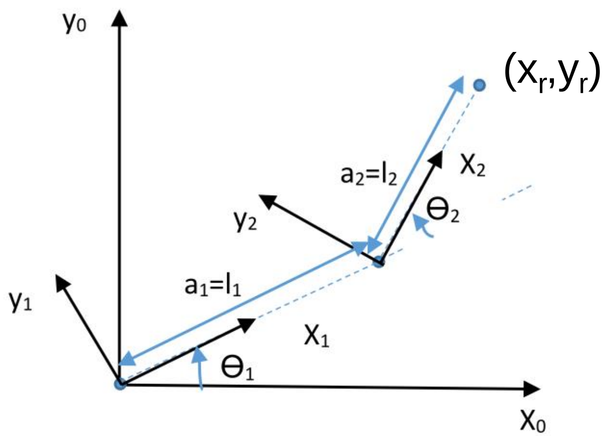
In what follows, we will describe two of them. First calculating the manipulator Jacobian matrix by **differentiating the forward kinematics equations** and second by **velocity propagation from link to link (geometric approach)**

Method 1: calculating the manipulator Jacobian matrix by differentiating the forward/direct kinematics equations of the mechanism

In many cases, we can directly compute the derivative of the robot tool pose w.r.t. the joint control variables from the forward kinematics equations to form the manipulator Jacobian.

This is straigthforward for linear velocity but can be applied to calculate the rotational velocity of the tool frame only in special cases like in the example on the following slide.

Example, planar RR-robot arm:



forward kinematics

$x_r = l_1 c_1 + l_2 c_{12}$
$y_r = l_1 s_1 + l_2 s_{12}$
$\phi_r = \theta_1 + \theta_2$

1.

first, take the derivative of both sides of the equations:

$\dot{x}_r = -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2)$
$\dot{y}_r = l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2)$
$\dot{\phi}_r = \omega_z = \dot{\theta}_1 + \dot{\theta}_2$

from which we can then get the manipulator Jacobian by regrouping terms on the right side of the equations into the Jacobian matrix and the joint velocity vector

2.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi}_r \end{bmatrix} = J_0(\theta) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Alternatively we could have just applyied the definition of Jacobian as the matrix of partial derivatives, so in the example we would have: $J_0 = \begin{bmatrix} \frac{dx_r}{d\theta_1} & \frac{dx_r}{d\theta_2} \\ \frac{dy_r}{d\theta_1} & \frac{dy_r}{d\theta_2} \\ \frac{d\phi_r}{d\theta_1} & \frac{d\phi_r}{d\theta_2} \end{bmatrix}$

and so we get for the Jacobian matrix

$$J_0(q) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ 1 & 1 \end{bmatrix}$$

Method 2: calculating the manipulator Jacobian matrix by velocity propagation from link to link (geometric approach)

For more details see **Craig's text book** pages 144-150.

Angular velocity of frame ***i*+1** expressed w.r.t frame ***i*+1**

Angular velocity of frame ***i*** expressed w.r.t frame ***i*+1**

Rotation velocity of joint i+1 around z_{i+1} axis

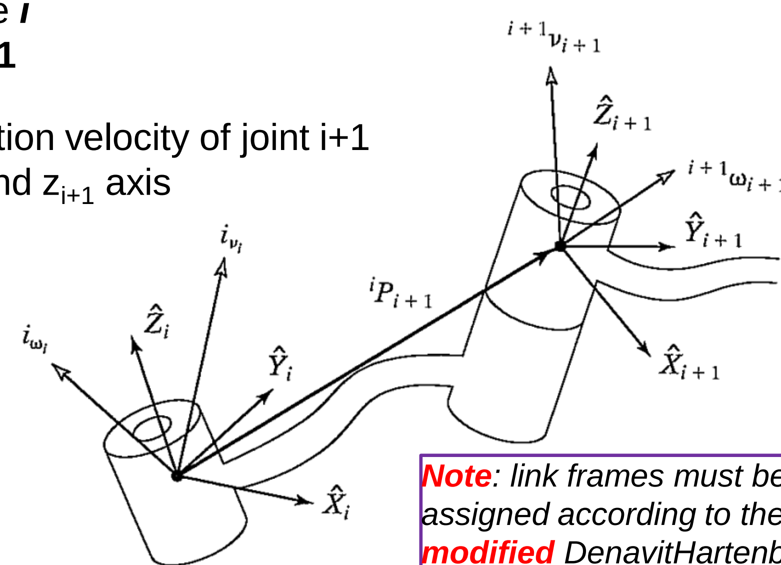
$$^{i+1}\omega_{i+1} = {}^{i+1}_{i}R\,{}^{i}\omega_i + \boxed{\dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1}} \quad \textbf{(5.45)}$$

Linear velocity of frame ***i*+1** expressed w.r.t frame ***i*+1**

Linear velocity of frame ***i*** expressed w.r.t frame ***i***

$$^{i+1}v_{i+1} = {}^{i+1}_{i}R({}^{i}v_i + {}^{i}\omega_i \times {}^{i}P_{i+1}) \quad \textbf{(5.47)}$$

Linear velocity due to a rotating vector



*Note: link frames must be assigned according to the **modified** DenavitHartenberg-convention*

FIGURE 5.7: Velocity vectors of neighboring links.

However, if joint i+1 is prismatic, we apply the equations:

for the rotational velocity: ${}^{i+1}\omega_{i+1} = {}^{i+1}\mathbf{R}_i \, {}^{i}\omega_i$

(5.48)

and for the linear velocity: ${}^{i+1}v_{i+1} = {}^{i+1}\mathbf{R}_i ({}^{i}v_i + {}^{i}\omega_i \times {}^{i}\mathbf{P}_{i+1}) + \dot{d}_{i+1} \, {}^{i+1}\hat{Z}_{i+1}$

Note that in the equations (5.45), (5.47) and (5.48) we have the rotations matrices ${}^{i+1}\mathbf{R}_i$ (or analogously ${}^{j}\mathbf{R}_{j-1}$), which can be acquired from the link transformations matrices formed by assigning the Modified Denavith Hartenberg parameters from each line of the parameter table into the symbolic form of the MDH-link transformation matrix:

$$ {}^{j-1}\mathbf{T}_j = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & a_{j-1} \\ s\theta_j c\alpha_{j-1} & c\theta_j c\alpha_{j-1} & -s\alpha_{j-1} & -s\alpha_{j-1} d_j \\ s\theta_j s\alpha_{j-1} & c\theta_j s\alpha_{j-1} & c\alpha_{j-1} & c\alpha_{j-1} d_j \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

And then by transposing the rotation part of the transformation matrix ${}^{j-1}\mathbf{T}_j$ to get ${}^{j}\mathbf{R}_{j-1}$ (or analogously ${}^{i}\mathbf{T}_{i+1}$ to get ${}^{i+1}\mathbf{R}_i$)

The calculation of the linear velocity based on a rotating vector has been illustrated in Craig’s text book Figure 5.5, page 140:

$$|\Delta Q| = (|^A Q| \sin\theta)(|^A \Omega_B|\Delta t) \quad (5.10)$$

$$^A V_Q = {}^A\Omega_B \times {}^A Q \quad (5.9)$$

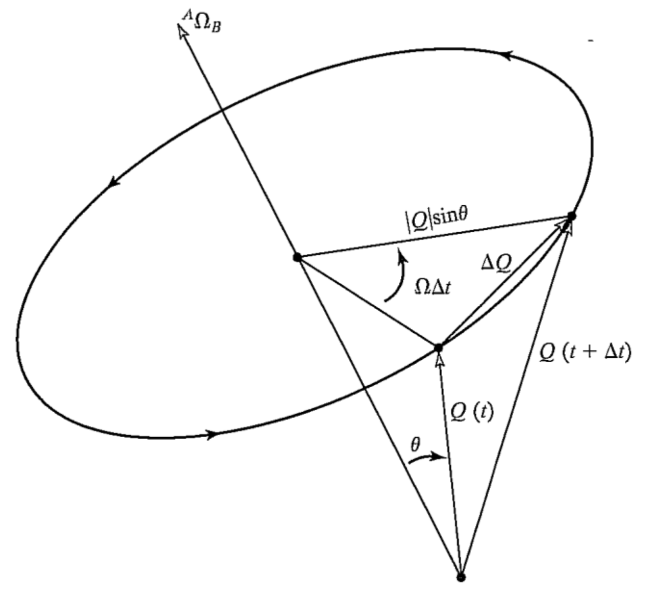


FIGURE 5.5: The velocity of a point due to an angular velocity.

An example from Craig's text book pp. 146-148

The aim is to calculate linear and angular velocities of the tip of the arm (frame 3) as a function of joint rates. We are supposed to express the velocity both w.r.t frame 3 as well as frame 0.

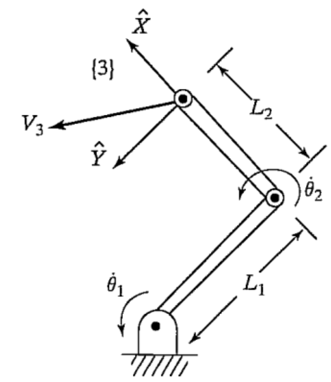


FIGURE 5.8: A two-link manipulator.

Note: link frames (1 and 2) have been assigned according to the ***modified*** *DenavitHartenberg-convention*

The ***R***-sub-matrices of the link matrices will be needed to project velocities between frames (*compare equations (5.45) and (5.47)*)

$$
{}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
{}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.49)
$$

Link matrices 1 and 2

$$
{}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

Tool transformation matrix

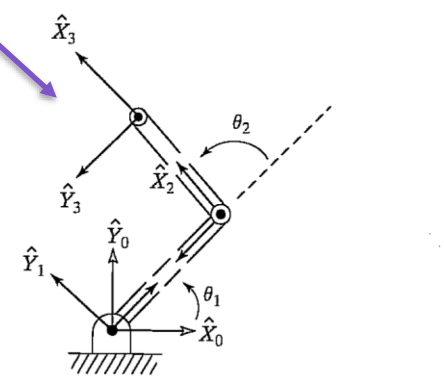


FIGURE 5.9: Frame assignments for the two-link manipulator.

By using equations (5.45) and (5.47) sequentially (*starting from the base of the robot*) we will propagate the velocities from link frame to link frame. Equations (5.54) and (5.55), <u>describing the angular and linear velocities of frame 3 expressed with respect to its own coordinate axes</u>, will be then the solution to the example problem.

Angular velocity of joint 1 around z_1 axis

Angular velocity propagated from frame 1 plus angular velocity of joint 2 around z_2 axis

No change in angular velocities between frames 2 and 3 because the frames are fixed to the same rigid link structure

$$ {}^{1}\omega_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \quad (5.50) $$

$$ {}^{2}\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \quad (5.52) $$

$$ {}^{3}\omega_3 = {}^{2}\omega_2 \quad (5.54) $$

$$ {}^{1}v_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.51) $$

$$ {}^{2}v_2 = \begin{bmatrix} c_2 & s_2 & 0 \\ -s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ l_1\dot{\theta}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 \\ 0 \end{bmatrix} \quad (5.53) $$

$$ {}^{3}v_3 = \begin{bmatrix} l_1 s_2 \dot{\theta}_1 \\ l_1 c_2 \dot{\theta}_1 + l_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \quad (5.55) $$

<u>Origin</u> of frame 1 does not move

Linear velocity caused by ${}^{1}\omega_1$ ”swinging" the ”shoulder” bar of length L_1 propagated to frame 2

Linear velocity of frame 2 plus linear velocity caused by ${}^{2}\omega_2$ ”swinging" the ”elbow” bar of length L_2 propagated to frame 3

However, if we needed to express the angular and linear velocities w.r.t the base frame (here 0-frame), we multiply the velocity vectors given by eq. (5.54) and (5.55) with the rotation matrix (5.56)

$$ {}^0_3R = {}^0_1R \;\; {}^1_2R \;\; {}^2_3R = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.56)$$

and so we get

$$ {}^0\omega_3 = {}^0_3\boldsymbol{R}\; {}^3\omega_3 = \begin{bmatrix} c_{12} & c_{12} & 0 \\ c_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{\theta}}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\boldsymbol{\theta}}_1 + \dot{\boldsymbol{\theta}}_2 \end{bmatrix}$$

$$ {}^0\upsilon_3 = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - l_2 s_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + l_2 c_{12}(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{bmatrix} \quad (5.57)$$

The angular velocities (*here around the z-axis only*) expressed w.r.t 0-frame and 3-frame are the same because the direction of the z-axis is the same among the two frames

Now we can form the Jacobian matrices by regrouping the terms, so for example Jacobian for the linear velocity becomes, (5.66) for calculating velocities which are expressed w.r.t tool frame (*i.e. frame 3*) and (5.67) w.r.t base frame (*i.e. frame 0*)

$$ {}^3J(\Theta) = \begin{bmatrix} l_1 s_2 & 0 \\ l_1 c_2 + l_2 & l_2 \end{bmatrix} \quad (5.66)$$

Note ${}^0\boldsymbol{J}(\boldsymbol{\theta})$ is the same as the first two rows of the Jacobian matrix, $\boldsymbol{J}_0(\boldsymbol{\theta})$ acquired in the example of slide 9

$$ {}^0J(\Theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \quad (5.67)$$

The 3rd line for the Jacobian matrix, corresponding to the angular velocity of the tool frame $\dot{\emptyset}$, can be formed by regrouping the terms in equation (5.52) yielding

$$\dot{\emptyset} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

And after adding the left vector of the product as the third line for the Jacobian matrix, Eq. (5.67), we get

$$^{0}J(\theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ 1 & 1 \end{bmatrix}$$

Which is the same matrix as what we got with the other approach *(i.e. differentiating the forward kinematics equations)* on slide 9

Note that we have here restricted the task space for the planar robot to $[x \quad y \quad \emptyset]$. If we would express the jacobian matrix w.r.t. a 3D task space it would get the form:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

Transforming velocities between coordinate frames

Consider two frames $\{A\}$ and $\{B\}$ related by $${}^{A}\boldsymbol{T}_{B} = \begin{pmatrix} {}^{A}\boldsymbol{R}_{B} & {}^{A}\boldsymbol{t}_{B} \\ 0 & 1 \end{pmatrix}$$

then the spatial velocity of a point with respect to frame {A} can be expressed relative to frame {B} by

$${}^{B}\nu = {}^{B}\boldsymbol{J}_{A}\, {}^{A}\nu$$

where the Jacobian

$${}^{B}\boldsymbol{J}_{A} = \boldsymbol{J}_{v}({}^{A}\boldsymbol{T}_{B}) = \begin{pmatrix} {}^{B}\boldsymbol{R}_{A} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & {}^{B}\boldsymbol{R}_{A} \end{pmatrix}$$

is a 6×6 matrix and a function of the relative orientation between the frames

Velocity is a free vector, i.e. it has no origin only direction and length. That is why for transforming velocities between coordinate frames we need only to adjust the orientation of the velocity vector to correspond to the relative orientation between the original and the new coordinate frame.

Recommended reading:

Peter Corke, Robotics, Vision and Control, Fundamental Algorithms in MATLAB, Second Edition, Springer, 2017, pages 229-233.

Craig, J.J, Introduction to Robotics: Mechanics and Control, Third Edition, Prentice Hall, 2005, pages 144-151