Robotics I

Lecture 15

Differential Kinematics of Open Chain, Jacobian

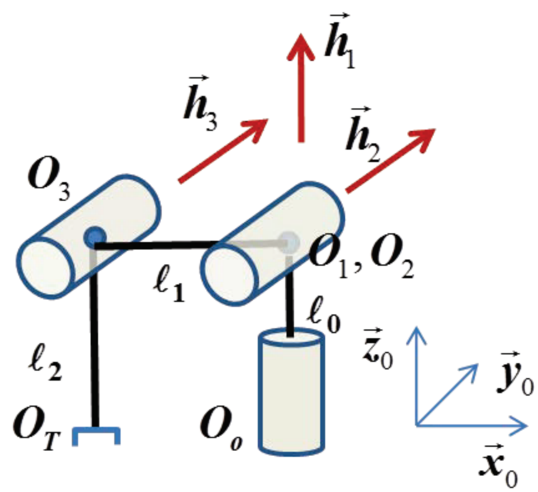
John T. Wen

October 24, 2011

Differential Kinematics

Differential kinematics: Mapping of joint velocities to spatial velocity (angular and linear velocities) of task frame.

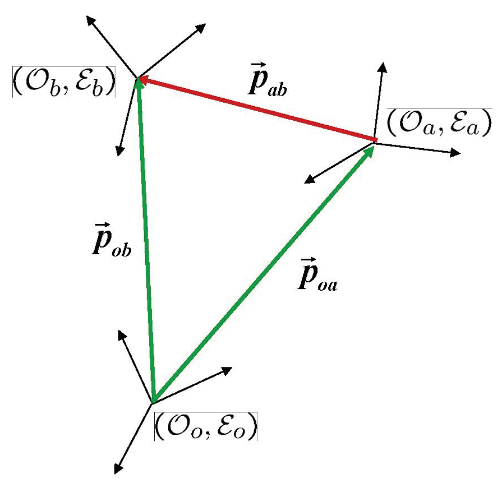
Example: Consider the Phantom Omni:



$$\dot{p}_{0T} = p_{01} + R_{01}p_{12} + R_{01}R_{12}p_{23} + R_{01}R_{12}p_{23}$$
$$\dot{p}_{0T} = p_{01} + \dot{R}_{01}p_{12} + \dot{R}_{01}R_{12}p_{23} + R_{01}\dot{R}_{12}p_{23} + \dot{R}_{01}R_{12}R_{23}p_{3T} + R_{01}\dot{R}_{12}R_{23}p_{3T} + R_{01}R_{12}\dot{R}_{23}p_{3T}.$$

Propagation of velocity in a rigidbody

Consider the propagation of spatial velocity in a rigidbody. Let A and B be two Euclidean frames attached to the same rigid body with spatial velocities $\begin{bmatrix} \vec{\omega}_A \\ \vec{v}_A \end{bmatrix}$ and $\begin{bmatrix} \vec{\omega}_B \\ \vec{v}_B \end{bmatrix}$ with respect to $(\mathcal{O}_o, \mathcal{E}_o)$.



Velocity Propagation in a Rigid Body (Vectorial)

Angular Vel.: $\mathcal{E}_A$ and $\mathcal{E}_B$ are on the same rigid body, so $\vec{\omega}_b = \vec{\omega}_a$.
Linear Vel.: $\vec{p}_{ob} = \vec{p}_{oa} + \vec{p}_{ab}$, $\vec{p}_{ab}$ is a constant vector in $\mathcal{E}_a$, so

$$\vec{v}_b = \frac{d\vec{p}_{ob}}{dt^0} = \frac{d\vec{p}_{oa}}{dt^0} + \vec{\omega}_a \times \vec{p}_{ab} = \vec{v}_a + \vec{\omega}_a \times \vec{p}_{ab}.$$

Stack up the angular velocity and linear velocity (note that $\vec{p}_{ba} = -\vec{p}_{ab}$):

$$\begin{bmatrix} \vec{\omega}_b \\ \vec{v}_b \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{I} & 0 \\ -\vec{p}_{ab}\times & \bar{I} \end{bmatrix}}_{\overline{\Phi}_{ba}} \begin{bmatrix} \vec{\omega}_a \\ \vec{v}_a \end{bmatrix}, \quad \begin{bmatrix} \vec{\omega}_a \\ \vec{v}_a \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{I} & 0 \\ \vec{p}_{ab}\times & \bar{I} \end{bmatrix}}_{\overline{\Phi}_{ab}=\overline{\Phi}_{ba}^{-1}} \begin{bmatrix} \vec{\omega}_b \\ \vec{v}_b \end{bmatrix}.$$

With $\{a,b\}$ corresponds to any two points fixed with respect to the rigid body, $\overline{\Phi}_{ab}$ forms a group:

$$\overline{\Phi}_{aa} = \bar{I},\ \overline{\Phi}_{ab} = \overline{\Phi}_{ba}^{-1},\ \overline{\Phi}_{ca} = \overline{\Phi}_{cb}\overline{\Phi}_{ba}.$$

Velocity Propagation in Coordinate Frame

Represent $(\vec{\omega}_b, \vec{v}_b)$ in $\mathcal{E}_b$ as (ω_b, v_b), and $(\vec{\omega}_a, \vec{v}_a)$ in $\mathcal{E}_a$ ad (ω_a, v_a):

$$\begin{bmatrix} \omega_b \\ v_b \end{bmatrix} = \underbrace{\begin{bmatrix} R_{ba} & 0 \\ -R_{ba}\widehat{p_{ab}} & R_{ba} \end{bmatrix}}_{:=\Phi_{ba}} \begin{bmatrix} \omega_a \\ v_a \end{bmatrix}.$$

We shall refer to $\vec{V}_a := \begin{bmatrix} \vec{\omega}_a \\ \vec{v}_a \end{bmatrix}$ as the (coordinate-independent) *spatial velocity* of the Euclidean frame $(O_a, \mathcal{E}_a)$. When it is represented in $\mathcal{E}_a$, we write it as $V_a := \begin{bmatrix} \omega_a \\ v_a \end{bmatrix}$.

Task Velocity due to Motion of Joint i

Consider the portion of arm from body i to arm tool frame. With all joints, $(q_{i+1}, \ldots, q_n)$, locked, this portion is a single rigid body. The the velocity of the ith joint may be propagated to the task velocity as:

If the ith joint is revolute: $\begin{bmatrix} \vec{\omega}_T \\ \vec{v}_T \end{bmatrix} = \begin{bmatrix} I & 0 \\ -\vec{p}_{iT} \times & I \end{bmatrix} \begin{bmatrix} \vec{h}_i \\ 0 \end{bmatrix} \dot{q}_i = \begin{bmatrix} \vec{h}_i \\ \vec{h}_i \times \vec{p}_{iT} \end{bmatrix} \dot{q}_i$

If the ith joint is prismatic: $\begin{bmatrix} \vec{\omega}_T \\ \vec{v}_T \end{bmatrix} = \begin{bmatrix} I & 0 \\ -\vec{p}_{iT} \times & I \end{bmatrix} \begin{bmatrix} 0 \\ \vec{h}_i \end{bmatrix} \dot{q}_i = \begin{bmatrix} 0 \\ \vec{h}_i \end{bmatrix} \dot{q}_i$

$\vec{p}_{iT} = \vec{p}_{i,i+1} + \vec{p}_{i+1,i+2} + \ldots + \vec{p}_{n,T}$.

If O_A is a *rigid* extension of O_T, the same expressions hold, with T replaced by A.

Jacobian

Putting all the joint velocities together we get (if all joints are revolute) and let $(O_A, \mathcal{E}_A)$ **be fixed with respect to the end effector:**

$$\begin{bmatrix} \vec{\omega}_A \\ \vec{v}_A \end{bmatrix} = J_A \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}, \quad J_A = \begin{bmatrix} \vec{h}_1 & \vec{h}_2 & \ldots & \vec{h}_n \\ \vec{h}_1 \times \vec{p}_{1A} & \vec{h}_2 \times \vec{p}_{2A} & \ldots & \vec{h}_n \times \vec{p}_{nA} \end{bmatrix}.$$

If the i**th joint is prismatic, then the** i**th column becomes** $\begin{bmatrix} 0 \\ \vec{h}_i \end{bmatrix}$.

For computation, we need to represent J_A **in a coordinate frame. For example, in the base frame,**

$$(J_A)_0 = \begin{bmatrix} h_1 & R_{01} h_2 & \ldots & R_{0,n-1} h_n \\ \widehat{h}_1 (p_{1A})_0 & \widehat{(h_2)_0} (p_{2A})_0 & \ldots & \widehat{(h_n)_0} (p_{nA})_0 \end{bmatrix}$$

where

$$(p_{iA})_0 = R_{0i} p_{i,i+1} + R_{0,i+1} p_{i+1,i+2} + \ldots + R_{0,n} p_{n,A}.$$

Decision #1: Choice of O_A

1. *Location of A frame (i.e., O_A).* **It doesn't have to be the physical end effector, it could be any rigid extension of the end effector, chosen to make J_A simple. Let's say the end effector tool frame is T frame and A is a rigid extension, i.e., $\mathcal{E}_T$ and $\mathcal{E}_A$ are the same, but the origin is related by $O_A = O_T + \vec{p}_{TA}$. Then the spatial velocities are related by**

$$\begin{bmatrix} \vec{\omega}_A \\ \vec{v}_A \end{bmatrix} = \begin{bmatrix} \bar{I} & 0 \\ -\vec{p}_{TA}\times & \bar{I} \end{bmatrix} \begin{bmatrix} \vec{\omega}_T \\ \vec{v}_T \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \vec{\omega}_T \\ \vec{v}_T \end{bmatrix} = \begin{bmatrix} \bar{I} & 0 \\ \vec{p}_{TA}\times & \bar{I} \end{bmatrix} \begin{bmatrix} \vec{\omega}_A \\ \vec{v}_A \end{bmatrix}.$$

Therefore, finding J_A as good as finding J_T: $\quad J_T = \begin{bmatrix} \bar{I} & 0 \\ \vec{p}_{TA}\times & \bar{I} \end{bmatrix} J_A.$

Typically, a good choice is the point of intersection of multiple rotational axes (so there are lots of zero vectors in J_A).

Decision #2: Coordinate frame to represent J_A

2. ***Choice of coordinate frame $\mathcal{E}_B$ to represent J_A.*** **This should be chosen to make $(J_A)_B$ simple (e.g., to make the coordinate transformation matrices within $(J_A)_B$ simple).**

Example

3-DOF Planar, SCARA, PUMA 560

Recall Iterative Solution Inverse Kinematics

Given $(p,R) \in SE(3)$. Let β be a 3-parameter representation of R. Define the end effector error as

$$e = \frac{1}{2}\|p_{0T} - p\|^2 + \frac{1}{2}\|\beta_{0T} - \beta\|^2 = w_1\frac{1}{2}(p_{0T} - p)^T(p_{0T} - p) + w_2\frac{1}{2}(\beta_{0T} - \beta)^T(\beta_{0T} - \beta),$$

where p_{0T} and β_{0T} are the position and orientation forward kinematics maps (which depend on q) and w_1 and w_2 are weighting factors.

The iterative inverse kinematics approach iteratively adjusts q to minimize e. If $e = 0$, *an* inverse kinematics solution is found.

Jacobian based Inverse Kinematics

The derivative of e **with respect to a parameter** t **is**

$$
\begin{aligned}
\frac{de}{dt} &= w_1(p_{0T}-p)^T\frac{dp_{0T}}{dt} + w_2(\beta_{0T}-\beta)^T\frac{d\beta_{0T}}{dt} \\
&= \begin{bmatrix} w_1(p_{0T}-p) \\ w_2(\beta(q)-\beta) \end{bmatrix}^T \begin{bmatrix} \frac{dp_{0T}}{dt} \\ \frac{d\beta_{0T}}{dt} \end{bmatrix} \\
&= \begin{bmatrix} w_1(p_{0T}-p) \\ w_2 J_\beta^T(\beta(q)-\beta) \end{bmatrix}^T \begin{bmatrix} \frac{dp_{0T}}{dt} \\ \omega \end{bmatrix} \\
&= \begin{bmatrix} w_1(p_{0T}-p) \\ w_2 J_\beta^T(\beta(q)-\beta) \end{bmatrix}^T (J_T)_o\frac{dq}{dt}.
\end{aligned}
$$

Jacobian based Inverse Kinematics

We can choose the joint coordinate update to be a gradient descent to reduce the error e:

$$\frac{dq}{dt} = -\alpha (J_T)_o^T \begin{bmatrix} w_1(p_{0T} - p) \\ w_2 J_\beta^T (\beta(q) - \beta) \end{bmatrix}^T .$$

The error e will continue to descrease as long as $(J_T)_o$ has full row rank and J_β is invertible.

If the arm is non-redundant (i.e., $(J_T)_o$ is square), then we could also use the Newton's descent:

$$\frac{dq}{dt} = -\alpha (J_T)_o^{-1} \begin{bmatrix} w_1(p_{0T} - p) \\ w_2 J_\beta^T (\beta(q) - \beta) \end{bmatrix}^T .$$

The algorithm also breaks down near the arm and/or representation singularities.

