Differential kinematics

• The goal of the differential kinematics is to find the relationship between the joint velocities and the end-effector linear and angular velocities

• Instantaneous velocity mappings can be obtained through time derivation of the direct kinematics function or geometrically at the differential level

• Different treatments arise for rotational quantities

• Establish the link between angular velocity and
  – time derivative of a rotation matrix
  – time derivative of the angles in a minimal representation of orientation

Geometric Jacobian

• it is desired to express the end-effector linear velocity $\dot{p}_e$ and angular velocity $\omega_e$ as a function of the joint velocities $\dot{q}$.

$$
\dot{p}_e = J_P(q)\dot{q}
$$

$$
\omega_e = J_O(q)\dot{q}.
$$

$$
\mathbf{v}_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J(q)\dot{q}
$$

$$
J = \begin{bmatrix} J_P \\ J_O \end{bmatrix},
$$

• The (6xn) matrix $J$ is the manipulator geometric Jacobian
Derivative of a Rotation Matrix

- Consider a time-varying rotation matrix $R = R(t)$. In view of the orthogonality of $R$
  \[ R(t) R^T(t) = I \]
  \[ \dot{R}(t) R^T(t) + R(t) \dot{R}^T(t) = 0. \]
  \[ S(t) = \dot{R}(t) R^T(t); \quad R(t) = S(t) R(t) \]

Consider a constant vector $p'$ and the vector $p(t) = R(t)p'$. The time derivative of $p(t)$ is
  \[ \dot{p}(t) = \dot{R}(t)p', \quad \dot{p}(t) = S(t) R(t)p' \]

If the vector $\omega(t) = [\omega_x, \omega_y, \omega_z]^T$ denotes the angular velocity of frame $R(t)$ with respect to the reference frame at time $t$, it is known from mechanics that
  \[ \dot{p}(t) = \omega(t) \times R(t)p' \]

Thus
  \[ S = \begin{bmatrix}
    0 & -\omega_z & \omega_y \\
    \omega_z & 0 & -\omega_x \\
    -\omega_y & \omega_x & 0
  \end{bmatrix} \]

and finally
  \[ \dot{R} = S(\omega) R \]

**Example:** Time derivative of an elementary rotation matrix

\[ R_x(\phi(t)) = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \phi(t) & -\sin \phi(t) \\
  0 & \sin \phi(t) & \cos \phi(t)
\end{bmatrix} \]

\[ \dot{R}_x(\phi) R^T_x(\phi) = \dot{\phi} \begin{bmatrix}
  0 & 0 & 0 \\
  0 & -\sin \phi & -\cos \phi \\
  0 & \cos \phi & -\sin \phi
\end{bmatrix} \begin{bmatrix}
  1 & 0 & 0 \\
  0 & \cos \phi & \sin \phi \\
  0 & -\sin \phi & \cos \phi
\end{bmatrix} \]

\[ \begin{bmatrix}
  0 & 0 & 0 \\
  0 & 0 & -\dot{\phi} \\
  0 & \dot{\phi} & 0
\end{bmatrix} = S(\omega) \]

\[ \omega = \begin{bmatrix}
  \dot{\phi} \\
  0 \\
  0
\end{bmatrix} \]
• Robot Jacobian matrices

• 1- **analytical** Jacobian (obtained by time differentiation)

\[
\begin{align*}
\mathbf{r} &= \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = f_r(q) & \mathbf{\dot{r}} &= \begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \frac{\partial f_r(q)}{\partial q} \, \dot{q} = J_r(q) \, \dot{q}
\end{align*}
\]

• 2- **geometric** Jacobian (no derivatives)

\[
\begin{align*}
\begin{bmatrix} \nu \\ \omega \end{bmatrix} &= \begin{bmatrix} \dot{p} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} J_{l}(q) \\ J_{\phi}(q) \end{bmatrix} \, \dot{q} = J(q) \, \dot{q}
\end{align*}
\]

• Analytical Jacobian of planar 2R arm

Direct kinematics:

\[
\begin{align*}
\begin{bmatrix} p_x \\ p_y \end{bmatrix} &= \begin{bmatrix} l_1 \, c_1 + l_2 \, c_{12} \\ l_1 \, s_1 + l_2 \, s_{12} \end{bmatrix} \\
\phi &= q_1 + q_2 \\
J_r(q) &= \begin{bmatrix} -l_1 \, s_1 - l_2 \, s_{12} & -l_2 \, s_{12} \\ l_1 \, c_1 + l_2 \, c_{12} & l_2 \, c_{12} \\ 1 & 1 \end{bmatrix}
\end{align*}
\]

Given \( r \), this is a 3 x 2 matrix.
Analytical Jacobian of polar robot

\[ \begin{align*}
\dot{p} = \dot{q} & = \begin{bmatrix} -q_3 c_2 s_1 & -q_3 s_2 c_1 & c_2 c_1 \\
q_3 c_2 c_1 & -q_3 s_2 s_1 & c_2 s_1 \\
0 & q_3 c_2 & s_2
\end{bmatrix} \begin{bmatrix} \dot{q}_1 \\
\dot{q}_2 \\
\dot{q}_3
\end{bmatrix} = J(q) \dot{q} \\
\frac{\partial f_r(q)}{\partial q}
\end{align*} \]

direct kinematics (here, \( r = p \))
\[ p_x = q_3 c_2 c_1 \\
p_y = q_3 c_2 s_1 \\
p_z = d_1 + q_3 s_2 \]

taking the time derivative

Geometric Jacobian

always a 6 x n matrix
\[ \begin{bmatrix} \dot{v}_E \\
\dot{\omega}_E
\end{bmatrix} = \begin{bmatrix} J_{l1}(q) & \cdots & J_{ln}(q) \\
J_{a1}(q) & \cdots & J_{an}(q)
\end{bmatrix} \begin{bmatrix} \dot{q}_1 \\
\vdots \\
\dot{q}_n
\end{bmatrix} \]

superposition of effects

\[ \begin{align*}
\dot{v}_E = J_{l1}(q) \dot{q}_1 + \cdots + J_{ln}(q) \dot{q}_n \\
\dot{\omega}_E = J_{a1}(q) \dot{q}_1 + \cdots + J_{an}(q) \dot{q}_n
\end{align*} \]

correlation to the linear e-e velocity due to \( \dot{q}_1 \)
correlation to the angular e-e velocity due to \( \dot{q}_1 \)

linear and angular velocity belong to (linear) vector spaces in \( \mathbb{R}^3 \)
• Contribution of revolute joint

\[
\begin{array}{|c|c|}
\hline
\text{revolute} & \text{i-th joint} \\
\hline
J_{li}(q) \dot{q}_i & (z_{i-1} \times p_{i-1,E}) \dot{\theta}_i \\
J_{ai}(q) \dot{q}_i & z_{i-1} \dot{\theta}_i \\
\hline
\end{array}
\]

• Contribution of prismatic joint

Note: joints beyond the i-th one are considered to be “frozen”, so that the distal part of the robot is a single rigid body.

\[
\begin{array}{|c|c|}
\hline
\text{prismatic} & \text{i-th joint} \\
\hline
J_{li}(q) \dot{q}_i & z_{i-1} \dot{d}_i \\
J_{ai}(q) \dot{d}_i & 0 \\
\hline
\end{array}
\]
• Expression of Geometric Jacobian

\[
\begin{pmatrix}
  \dot{p}_{0,E} \\
  \dot{\omega}_E
\end{pmatrix} =
\begin{pmatrix}
  v_E \\
  \omega_E
\end{pmatrix} =
\begin{pmatrix}
  J_l(q) \\
  J_M(q)
\end{pmatrix} \dot{q} =
\begin{pmatrix}
  J_{l1}(q) & \cdots & J_{ln}(q) \\
  J_{M1}(q) & \cdots & J_{Mn}(q)
\end{pmatrix}
\begin{pmatrix}
  \dot{q}_1 \\
  \vdots \\
  \dot{q}_n
\end{pmatrix}
\]

<table>
<thead>
<tr>
<th>prismatic</th>
<th>revolute</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-th joint</td>
<td>i-th joint</td>
</tr>
<tr>
<td>(J_l(q))</td>
<td>(Z_{i-1} \times P_{i-1,E})</td>
</tr>
<tr>
<td>(J_M(q))</td>
<td>(0)</td>
</tr>
</tbody>
</table>

This can be also computed as:

\[
\frac{\partial p_{0,E}}{\partial q_i} = \begin{pmatrix}0 & 0 & 1 \end{pmatrix}
\]

\[
z_{i-1} = ^0R_1(q_1)\cdots^1R_{i-1}(q_{i-1})\begin{pmatrix}0 \\
0 \\
1\end{pmatrix}
\]

\[
P_{i-1,E} = p_{0,E}(q_1,\ldots,q_n) - p_{0,i-1}(q_1,\ldots,q_{i-1})
\]

All vectors should be expressed in the same reference frame (here, the base frame \(RF_0\)).

• Geometric Jacobian of planar 2R arm

\[
z_0 = z_1 = z_2 = \begin{pmatrix}0 \\
0 \\
1\end{pmatrix}
\]

\[
A_1 = \begin{pmatrix}c_1 & -s_1 & 0 \\
s_1 & c_1 & 0 \\
0 & 0 & 1\end{pmatrix}
\]

\[
P_{0,1} = \begin{pmatrix}l_1c_1 \\
l_1s_1 \\
0 \end{pmatrix}
\]

\[
P_{i,E} = p_{0,E} - p_{0,1}
\]

\[
A_2 = \begin{pmatrix}c_{12} & -s_{12} & 0 \\
s_{12} & c_{12} & 0 \\
0 & 0 & 1 \end{pmatrix}
\]

\[
P_{0,E} = \begin{pmatrix}l_1c_{12} + l_2c_{12} \\
l_1s_{12} + l_2s_{12} \\
0 \end{pmatrix}
\]
Jacobian of Three-link Planar Arm

\[
\mathbf{J}(\mathbf{q}) = \begin{bmatrix}
\mathbf{z}_0 \times (\mathbf{p}_3 - \mathbf{p}_0) \\
\mathbf{z}_1 \times (\mathbf{p}_3 - \mathbf{p}_1) \\
\mathbf{z}_2 \times (\mathbf{p}_3 - \mathbf{p}_2)
\end{bmatrix}
\]

\[
\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{p}_1 = \begin{bmatrix} a_1s_1 \\ a_1c_1 \\ 0 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} a_1s_1 + a_2s_12 \\ a_1c_1 + a_2c_12 \\ 0 \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} a_1s_1 + a_2s_12 + a_3s_123 \\ a_1c_1 + a_2c_12 + a_3c_123 \\ 0 \end{bmatrix}
\]

\[
\mathbf{z}_0 = \mathbf{z}_1 = \mathbf{z}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]

\[
\mathbf{J} = \begin{bmatrix}
-a_1s_1 - a_2s_12 - a_3s_123 \\ a_1c_1 + a_2c_12 + a_3c_123 \\ 0 \\
-a_2s_12 - a_3s_123 \\ -a_2c_12 - a_3c_123 \\ 0 \\
-a_3s_123 \\ -a_3c_123 \\ 0 \end{bmatrix}
\]
• If orientation is of no concern, the \((2\times3)\) Jacobian for the positional part can be derived by considering just the first two rows, i.e.,

\[
J_p = \begin{bmatrix}
-a_1 s_1 - a_2 s_{12} - a_3 s_{123} & -a_2 s_{12} - a_3 s_{123} & -a_3 s_{123} \\
-a_1 c_1 + a_2 c_{12} + a_3 c_{123} & -a_2 c_{12} + a_3 c_{123} & a_3 c_{123}
\end{bmatrix}
\]

• Jacobian of Anthropomorphic Arm

\[
J = \begin{bmatrix}
z_0 \times (p_3 - p_0) & z_1 \times (p_3 - p_1) & z_2 \times (p_3 - p_2)
\end{bmatrix}
\]

\[
p_0 = p_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad p_2 = \begin{bmatrix} a_2 c_1 c_2 \\ a_2 s_1 c_2 \\ a_2 s_2 \end{bmatrix}, \quad p_3 = \begin{bmatrix} c_1 (a_2 c_2 + a_3 c_{23}) \\ s_1 (a_2 c_2 + a_3 c_{23}) \\ a_2 s_2 + a_3 s_{23} \end{bmatrix}
\]

\[
z_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad z_1 = z_2 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}
\]

\[
J = \begin{bmatrix}
-s_1 (a_2 c_2 + a_3 c_{23}) & -c_1 (a_2 s_2 + a_3 s_{23}) & -a_3 c_1 s_{23} \\
c_1 (a_2 c_2 + a_3 c_{23}) & -s_1 (a_2 s_2 + a_3 s_{23}) & -a_3 s_1 s_{23} \\
0 & a_2 c_2 + a_3 c_{23} & a_3 c_{23}
\end{bmatrix}
\]

\[
0 \\ 0 \\ s_1 \\
0 \\ -c_1 \\ -c_1 \\
1 \\ 0 \\ 0
\]
- Notice that the Jacobian matrix depends on the frame in which the end-effector velocity is expressed.
- If it is desired to represent the Jacobian in a different frame $u$, it is sufficient to know the relative rotation matrix $Ru$.

\[
\begin{bmatrix}
\dot{p}_e^u \\
\omega_e^u
\end{bmatrix} = \begin{bmatrix}
R^u & O \\
O & R^u
\end{bmatrix} \begin{bmatrix}
\dot{p}_e \\
\omega_e
\end{bmatrix},
\]

\[
\begin{bmatrix}
\dot{p}_e^u \\
\omega_e^u
\end{bmatrix} = \begin{bmatrix}
R^u & O \\
O & R^u
\end{bmatrix} J \dot{q}
\]

\[
J^u = \begin{bmatrix}
R^u & O \\
O & R^u
\end{bmatrix} J,
\]

- where $J^u$ denotes the geometric Jacobian in Frame $u$.

- Having $^0J_n(q)$, what is $^BJ_E(q)$? (E is the endtip of the tools grasped tightly be EEF)

\[
\begin{bmatrix}
0_v \\
0_\omega
\end{bmatrix} = ^0J_n(q) \dot{q}
\]

\[
\begin{bmatrix}
^Bv_E \\
^B\omega
\end{bmatrix} = \begin{bmatrix}
^Bv_0 \\
0
\end{bmatrix} + \begin{bmatrix}
^Bv_0(q) \\
0
\end{bmatrix} \begin{bmatrix}
I & S(q_{En}) \\
0 & I
\end{bmatrix} \begin{bmatrix}
0_v \\
0_\omega
\end{bmatrix}
\]

\[
= \begin{bmatrix}
^Bv_0(q) \\
0
\end{bmatrix} + \begin{bmatrix}
^Bv_0(q) \\
0
\end{bmatrix} \begin{bmatrix}
I & S(q_{En}(q)) \\
0 & I
\end{bmatrix} ^0J_n(q) \dot{q} = ^BJ_E(q) \dot{q}
\]

never singular!
Kinematic Singularities

- configurations where the Jacobian loses rank
  - loss of instantaneous mobility of the robot end-effector
- for $m=n$, they correspond in general to Cartesian poses that lead to a number of inverse kinematic solutions that differs from the “generic” case
- “in” a singular configuration, one cannot find a joint velocity that realizes a desired end-effector velocity in an arbitrary direction of the task space
- “close” to a singularity, large joint velocities may be needed to realize some (even small) velocity of the end-effector
- finding and analyzing in advance all singularities of a robot helps in avoiding them during trajectory planning and motion control
  - when $m=n$: find the configurations $q$ such that $\det J(q) = 0$
  - when $m<n$: find the configurations $q$ such that all $m \times m$ minors of $J$ are singular (or, equivalently, such that $\det [J(q) J^T(q)] = 0$)
- finding all singular configurations of a robot with a large number of joints, or the actual “distance” from a singularity, is a hard computational task
• Singularity of Planar 2R arm

![Diagram of a planar 2R arm]

For manipulators having a spherical wrist, by analogy with what has been seen for IK, it is possible to split the problem of singularity computation into two separate problems:

- computation of arm singularities resulting from the motion of the first 3 or more links,
- computation of wrist singularities resulting from the motion of the wrist joints.

Consider the case n=6. The Jacobian can be partitioned into (3 x 3) blocks:

\[ J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \]

It can be shown that (show yourself)

\[ \text{det}(J) = \text{det}(J_{11}) \text{det}(J_{22}) \]

and thus, \( \text{det}(J)=0 \) leads to

\[ \text{det}(J_{11}) = 0 \quad \text{(arm singularities)} \quad \text{det}(J_{22}) = 0 \quad \text{(wrist singularities)} \]
• Arm Singularities
• For anthropomorphic arm Jacobian for the linear velocity part is given by

\[
J_p = \begin{bmatrix}
-s_1(a_2c_2 + a_3c_23) & -c_1(a_2s_2 + a_3s_23) & -a_3c_1s_23 \\
-c_1(a_2c_2 + a_3c_23) & -s_1(a_2s_2 + a_3s_23) & -a_3s_1s_23 \\
0 & a_2c_2 + a_3c_23 & a_3c_23
\end{bmatrix}
\]

\[
\det(J_p) = -a_2a_3s_3(a_2c_2 + a_3c_23).
\]
• The determinant vanishes if

Fig. 3.5. Anthropomorphic arm at an elbow singularity
\[\dot{\vartheta}_3 = 0 \quad \dot{\vartheta}_3 = \pi\]

Fig. 3.6. Anthropomorphic arm at a shoulder singularity
\[p_x = p_y = 0\]

• Wrist Singularities

\[
J_{22} = \begin{bmatrix}
z_3 & z_4 & z_5
\end{bmatrix}.
\]

• wrist is at a singular configuration whenever the unit vectors \(z_3, z_4, z_5\) are linearly dependent. This occurs when \(z_3\) and \(z_5\) are aligned

Fig. 3.4. Spherical wrist at a singularity
\[\dot{\vartheta}_5 = 0 \quad \dot{\vartheta}_5 = \pi\]
Analytical Jacobian

- Remember that it is possible to describe the end-effector pose by means of the \((m \times 1)\) vector, with \(m \leq n\)

\[
x_e = \begin{bmatrix} p_e \\ \phi_e \end{bmatrix}
\]

Thus

\[
\dot{p}_e = \frac{\partial p_e}{\partial q} \dot{q} = J_P(q) \dot{q}.
\]

\[
\dot{\phi}_e = \frac{\partial \phi_e}{\partial q} \dot{q} = J_\phi(q) \dot{q}.
\]

\[
\dot{x}_e = \begin{bmatrix} \dot{p}_e \\ \dot{\phi}_e \end{bmatrix} = \begin{bmatrix} J_P(q) \\ J_\phi(q) \end{bmatrix} \dot{q} = J_A(q) \dot{q}
\]

- the relationship between the angular velocity and the derivative for a given set of Euler angles ZYZ

\[
\omega_e = \dot{\hat{\phi}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \dot{\hat{\psi}} \begin{bmatrix} -s_\phi \\ c_\phi s_\theta \\ c_\phi c_\theta \end{bmatrix} + \dot{\hat{\theta}} \begin{bmatrix} c_\phi s_\theta \\ s_\phi s_\theta \\ s_\phi c_\theta \end{bmatrix}
\]

\[
\omega_e = T(\phi_e) \dot{\phi}_e,
\]

\[
T = \begin{bmatrix} 0 & -s_\phi & c_\phi s_\theta \\ 0 & c_\phi & s_\phi s_\theta \\ 1 & 0 & c_\theta \end{bmatrix}
\]
• Once the transformation $T$ is given, the analytical Jacobian can be related to the geometric Jacobian as

$$v_e = \begin{bmatrix} I & O \\ O & T(\phi_e) \end{bmatrix} \dot{x}_e = T_A(\phi_e)\dot{x}_c$$

$$J = T_A(\phi)J_A$$
Inverse Differential Kinematics

- The differential kinematics equation represents a linear mapping between the joint velocity space and the operational velocity space.

- Suppose that a motion trajectory is assigned to the end-effector in terms of $v_e$ and the initial conditions on position and orientation. The aim is to determine a feasible joint trajectory $(q(t), \dot{q}(t))$ that reproduces the given trajectory.

- Considering $n = m$, the joint velocities can be obtained via simple inversion of the Jacobian matrix

  $$
  \dot{q} = J^{-1}(q)v_e
  $$

- If the initial manipulator posture $q(0)$ is known, joint positions can be computed by integrating velocities over time, i.e.,

  $$
  q(t) = \int_0^t \dot{q}(\zeta)d\zeta + q(0)
  $$

- The integration can be performed in discrete time by resorting to numerical techniques. The simplest technique is based on the Euler integration method

  $$
  q(t_{k+1}) = q(t_k) + \dot{q}(t_k)\Delta t.
  $$

- This technique for inverting kinematics is independent of the solvability of the kinematic structure. Nonetheless, it is necessary that the Jacobian be square and of full rank; this demands further insight into the cases of redundant manipulators and kinematic singularity occurrence.
• **IK Algorithm: Jacobian (pseudo-)inverse**

  The *operational space error* between the desired and the actual end-effector position and orientation

  \[ e = \mathbf{x}_d - \mathbf{x}_e \]

  • On the assumption that matrix \( J_A \) is square and nonsingular, the choice

  \[ \dot{\mathbf{q}} = J_A^{-1}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}e) \]

  leads to the equivalent linear system

  \[ \dot{\mathbf{e}} + \mathbf{K}\mathbf{e} = 0 \]

  • If \( \mathbf{K} \) is a positive definite (usually diagonal) matrix, the system is *asymptotically stable*. The error tends to zero along the trajectory with a convergence rate that depends on the eigenvalues of matrix \( \mathbf{K} \).

  for a constant reference (\( \dot{\mathbf{x}}_d = 0 \)) this methods gives the configuration associated with given task space position
• Behavior near a singularity

- find the joint velocity vector that realizes a desired end-effector "generalized" velocity (linear and angular)

\[ \mathbf{v} = J(\mathbf{q}) \dot{\mathbf{q}} \quad \text{non-singular} \quad \dot{\mathbf{q}} = J^{-1}(\mathbf{q}) \mathbf{v} \]

- problems
  - near a singularity of the Jacobian matrix (high \( \dot{\mathbf{q}} \))
  - for redundant robots (no standard "inverse" of a rectangular matrix)

  **in these cases, "more robust" inversion methods are needed**

---

problems arise only when commanding joint motion by inversion of a given Cartesian motion task

- figure shows a linear Cartesian trajectory for a planar 2R robot
- there is a sudden increase of the displacement/velocity of the first joint near \( \theta_2 = -\pi \) (end-effector close to the origin), despite the required Cartesian displacement is small
\[ \dot{q} = J^{-1}(q) \nu \]

regular case

A line from right to left, at \( \alpha = 170^\circ \) angle with x-axis, executed at constant speed \( \nu = 0.6 \text{ m/s} \) for \( T = 6 \text{ s} \)
\[ \dot{q} = J^{-1}(q) \nu \]

close to singular case

A line from right to left, at \( \alpha = 178^\circ \) angle with x-axis, executed at constant speed \( \nu = 0.6 \text{ m/s} \) for \( T = 6 \text{ s} \)
• Damped Least Squares method
• An alternative solution overcoming the problem of inverting differential kinematics in the neighborhood of a singularity is provided by the so-called damped least-squares (DLS) inverse

\[ \mathbf{J}^+ = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T + k^2 \mathbf{I})^{-1} \]

• Which minimize following objective function

\[
\min_{\dot{\mathbf{q}}} H = \frac{\lambda}{2} \| \dot{\mathbf{q}} \|^2 + \frac{1}{2} \| \mathbf{J} \dot{\mathbf{q}} - \mathbf{v} \|^2, \quad \lambda \geq 0
\]

- inversion of differential kinematics as an optimization problem
- function \( H \) = weighted sum of two objectives (minimum error norm on achieved end-effector velocity and minimum norm of joint velocity)
- \( \lambda = 0 \) when “far enough” from a singularity
- \( \mathbf{J}_{\text{DLS}} \) can be used both for \( m = n \) and for \( m < n \)

\[ \dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \mathbf{v} \quad \text{path at} \quad \alpha = 179.5^\circ \quad \dot{\mathbf{q}} = \mathbf{J}_{\text{DLS}}(\mathbf{q}) \mathbf{v} \]

here, a very fast reconfiguration of first joint ...

a completely different inverse solution, around/after crossing the region close to the folded singularity
Based on the given diagram and text, the Jacobian of the 2R arm with \( l_1 = l_2 = 1 \) and \( q_2 = 0 \) (rank \( \rho = 1 \)) is:

\[
J = \begin{bmatrix}
-2s_1 & -s_1 \\
2c_1 & c_1
\end{bmatrix}, \quad J^\dagger = \frac{1}{5} \begin{bmatrix}
-2s_1 & 2c_1 \\
-s_1 & c_1
\end{bmatrix}
\]

The joint velocity \( \dot{q} = J^\dagger v \) is the minimum norm joint velocity vector that realizes \( v \perp \mathcal{R}(J) \) and \( v \in \mathcal{N}(J^T) \).
Simulation of Inverse Kinematics Algorithms

Consider the 3-link planar arm in whose link lengths are

\[ a_1 = a_2 = a_3 = 0.5 \text{ m} \]

The initial posture \( q = [\pi, -\pi/2, -\pi/2]^T \text{ rad} \) correspond to the end-effector pose:

\[ p = [0, 0.5]^T \text{ m}, \phi = 0 \text{ rad} \]

A circular path of radius 0.25m and center at (0.25, 0.5)m is assigned to the end-effector. Let the motion desired trajectory be

\[
p_d(t) = \begin{bmatrix} 0.25(1 - \cos \pi t) \\ 0.25(2 + \sin \pi t) \end{bmatrix} \quad 0 \leq t \leq 4; \]

\[
\phi_d(t) = \sin \frac{\pi}{24} t \quad 0 \leq t \leq 4; \]

The inverse kinematics algorithms were implemented on a computer by adopting the Euler numerical integration scheme (3.48) with an integration time \( \Delta t = 1\text{ms} \).

At first, the inverse kinematics along the given trajectory has been performed by using (3.47).
Open loop inverse kinematics

Closed loop inverse kinematics

```
function ds = desire(t)
% desired trajectory
% 
% a_des = (0.25*(1-cos(pi*3*t]));
% 0.25*(7*sin(pi*3*t));
% sin(pi*(t+4*3))/4;
% (pi*cos(pi*(t+4*3))/24;
% }

% 
% if t<=
% a_des = [0; 0.5; 0.5];
% x_des = [0; 0.5];
% end
% ds = [x_des; a_des];

function [err_p, err_q] = FF(q, k_des)
% This block supports the Embedded MATLAB subset.
% See the help menu for details.

a1 = 5;
q1 = q(1);
q2 = q(2);
q3 = q(3);

%
% s = [a1*cos(q1)+a2*cos(q1-q2)+a3*cos(q1-q2-q3);
% a1*sin(q1)+a2*sin(q1-q2)+a3*sin(q1-q2-q3);
% q1+q2]
%
% J = [-a1sin(q1)-a2sin(q1-q2)-a3sin(q1-q2-q3),
% a1cos(q1)+a2cos(q1-q2)+a3cos(q1-q2-q3),
% a1cos(q1)+a2cos(q1-q2)+a3cos(q1-q2-q3),
% a2cos(q1-q2)+a3cos(q1-q2-q3),
% a3cos(q1-q2-q3);]
%
% dq = inv(J)*xd_des;
% k = [500, 0.0, 0.0, 0.0, 0.0, 0.0, 0.05];
% dq = inv(J)*k*(x_des-x);
```
Fig. 3.15. Time history of the norm of end-effector position error and orientation error with the open-loop inverse Jacobian algorithm

Fig. 3.16. Time history of the joint positions and velocities, and of the norm of end-effector position error and orientation error with the closed-loop inverse Jacobian algorithm
• **Statics: Transformation of forces**

  • The goal of *statics analysis* is to determine the relationship between the generalized forces applied to the end-effector and the generalized forces applied to the joints with the manipulator at an equilibrium configuration.

  • Let $\mathbf{\tau}$ denote the $(n \times 1)$ vector of joint torques and $\mathbf{y}$ the $(r \times 1)$ vector of end-effector forces.

  • The application of the *principle of virtual work* allows the determination of the required relationship as

  \[ \mathbf{\tau} = J^T(q)\mathbf{y}_e \]

  • It is worth remarking that the end-effector forces $\mathbf{y}_e \in N(J^T)$ are entirely absorbed by the structure in that the mechanical constraint reaction forces can balance them exactly.

  ![Diagram of a robot with forces](image.png)

  - $\mathbf{\tau}$ = vector of forces/torques at the joints
  - $\mathbf{F}$ = vector of forces/torques exerted from the robot end-effector to the environment (equal and opposite to the reaction forces/torques from the environment to the robot end-effector)
  - **which is the relation between $\mathbf{\tau}$ and $\mathbf{F}$ in a static equilibrium (i.e., with the robot remaining at rest)?
• Equivalent formulations

![Diagram of a robotic arm with torques and forces]

- which $\tau$ should be provided by the motors at the joints so that the robot end-effector applies $F$ to the environment?
- which $\tau$ at the joints balances a $-F$ exerted from the environment?
- which is the force/torque $\tau$ "felt" at the joints in the presence of a $-F$ exerted at the robot end-effector?

• Virtual displacements and works

![Diagram of virtual displacements with forces and moments]

- infinitesimal (or "virtual", i.e., satisfying all possible constraints imposed on the system) displacements at an equilibrium

- without kinetic energy variation (zero acceleration)
- without dissipative effects (zero velocity)

the "virtual work" is the work done by all forces/torques acting on the system for a given virtual displacement
• Principle of virtual work

The sum of the “virtual works” done by all external forces/torques acting on a system which is in equilibrium is zero.

\[ \tau^T dq - F^T \left[ \begin{array}{c} \frac{dp}{dt} \\ \omega dt \end{array} \right] = \tau^T dq - F^T J dq = 0 \quad \forall dq \]

\[ \tau = J^T(q) F \]

• Duality between velocity and force

The singular configurations for the velocity map are the same as those for the force map.

\[ \rho(J) = \rho(J^T) \]
Example of singularity analysis

Planar 2R arm with generic link lengths $l_1$ and $l_2$

$$J(q) = \begin{bmatrix} -l_1 s_1 l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \quad \det J(q) = l_1 l_2 s_2$$

Singularity at $q_2 = 0$ (arm straight)

$$\mathcal{N}(J) = \alpha \begin{bmatrix} -s_1 \\ c_1 \end{bmatrix} \quad \mathcal{N}(J^T) = \alpha \begin{bmatrix} c_1 \\ s_1 \end{bmatrix}$$

$$\mathcal{N}(J^T) = \beta \begin{bmatrix} l_1 + l_2 \\ l_2 \end{bmatrix} \quad \mathcal{N}(J) = \beta \begin{bmatrix} l_2 \\ -(l_1 + l_2) \end{bmatrix}$$

Singularity at $q_2 = \pi$ (arm folded)

$$J = \begin{bmatrix} -(l_1 + l_2) s_1 & -l_2 s_1 \\ (l_1 + l_2) c_1 & l_2 c_1 \end{bmatrix}$$

Velocity Manipulability Ellipsoid

- In a given configuration, we wish to evaluate how “effective” is the mechanical transformation between joint velocities and end-effector velocities
  - “how easily” can the end-effector be moved in the various directions of the task space
  - equivalently, “how far” is the robot from a singular condition
- We consider all end-effector velocities that can be obtained by choosing joint velocity vectors of unit norm

$$\dot{q}^T \dot{q} = 1$$

$$v^T J^\# v = 1$$

Rem: the “core” matrix of the ellipsoid equation $v^T A^{-1} v = 1$ is the matrix $A$!
planar 2R arm with unitary links

**length of principal (semi-)axes:**
singular values of $J$ (in its SVD)

$$\sigma_i \{J\} = \sqrt{\lambda_i \{JJ^T\}} \geq 0$$

in a singularity, the ellipsoid loses a dimension
(for $m=2$, it becomes a segment)

**direction of principal axes:**
(orthogonal) eigenvectors associated to $\lambda_i$

$$w = \sqrt{\text{det}JJ^T} = \prod_{i=1}^{m} \sigma_i \geq 0$$

proportional to the volume of the ellipsoid (for $m=2$, to its area)

---

planar 2R arm with unitary links: Jacobian $J$ is square

$$\sqrt{\text{det}(JJ^T)} = \sqrt{\text{det}J \cdot \text{det}J^T} = |\det J| = \prod_{i=1}^{m} \sigma_i$$

max at $\theta_2 = \pi/2$

**full isotropy is never obtained in this case, since it always $\sigma_1 \neq \sigma_2$**

$\sigma_1(J)$

$\sigma_2(J)$

max at $r=\sqrt{2}$

**best posture for manipulation (similar to a human arm)**
• Force manipulability
  - in a given configuration, evaluate how “effective” is the mechanical transformation between joint torques and end-effector forces
  - “how easily” can the end-effector apply generalized forces (or balance applied ones) in the various directions of the task space
  - in particular, in a singular configuration, there are directions in the task space where an external force/torque is balanced by the robot without the need of any joint torque
  - we consider all end-effector forces that can be applied (or balanced) by choosing joint torque vectors of unit norm
  \[ \tau^T \tau = 1 \]

  same directions of the principal axes of the velocity ellipsoid, but with semi-axes of inverse lengths

  task force manipulability ellipsoid

• Velocity and force manipulability: a dual comparison

  ![Planar 2R arm with unitary links](image)

  Cartesian actuation task (high joint-to-task transformation ratio): preferred velocity (or force) directions are those where the ellipsoid stretches

  Cartesian control task (low transformation ratio = high resolution): preferred velocity (or force) directions are those where the ellipsoid shrinks