Control of industrial robots

Review of robot kinematics

Prof. Paolo Rocco (paolo.rocco@polimi.it)
Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria
Introduction

- With these slides we will cover basic elements in robot kinematics.

- We will start from a basic problem of representation of a rigid body in space, and then proceed through the formal tools used in robotics till the definition of the direct, inverse and differential kinematics of the manipulator.

- All this material is well covered with better detail in any introductory Robotics course at BSc level. It is reviewed here for the sole purpose of making this course self-contained for students who lack this background.

Most of the pictures in these slides are taken from the textbook:

B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo: *Robotics: Modelling, Planning and Control, 3rd Ed.*
Springer, 2009
Let us consider a rigid body in space:

How can we characterize the position and orientation of the body in space?

The study of kinematics of mechanical bodies is facilitated if Cartesian frames are introduced. Each point in space has 3 coordinates \((x, y, z)\) in the Cartesian frame.
The best thing to do is to consider a reference frame and to attach a second frame to the body.

The problem is now how to characterize the position and orientation of a frame with respect to another one.
Position and orientation of a body in space

The representation of the position is just made with the components of the origin of the body-attached frame with respect to the reference frame:

\[
\begin{bmatrix}
O'_x \\
O'_y \\
O'_z
\end{bmatrix}
\]

The three components can be conveniently gathered in a vector:

\[
O' = \begin{bmatrix}
o'_x \\
o'_y \\
o'_z
\end{bmatrix}
\]
Position and orientation of a body in space

The representation of the orientation can be made considering unit length vectors along the axes of the rotated frame and evaluating their components in the reference frame:

We obtain three vectors:

\[
\mathbf{x}' = \begin{bmatrix} x'_x \\ x'_y \\ x'_z \end{bmatrix}, \quad \mathbf{y}' = \begin{bmatrix} y'_x \\ y'_y \\ y'_z \end{bmatrix}, \quad \mathbf{z}' = \begin{bmatrix} z'_x \\ z'_y \\ z'_z \end{bmatrix}
\]
We can gather the elements of $x', y', z'$ in a matrix:

$$R = [x' \ y' \ z'] = \begin{bmatrix} x'_x & y'_x & z'_x \\ x'_y & y'_y & z'_y \\ x'_z & y'_z & z'_z \end{bmatrix} = \begin{bmatrix} x'^T x & y'^T x & z'^T x \\ x'^T y & y'^T y & z'^T y \\ x'^T z & y'^T z & z'^T z \end{bmatrix}$$

This matrix is called **rotation matrix** of the frame $(x', y', z')$ with respect to the frame $(x, y, z)$.

Since the following relations hold:

$$x'^T x' = 1, \quad y'^T y' = 1, \quad z'^T z' = 1$$
$$x'^T y' = 0, \quad y'^T z' = 0, \quad z'^T x' = 0$$

we have: $R^T R = I$ (orthogonal matrix)
Elementary rotations

Let us consider a rotation by an angle $\alpha$ around $z$ axis:

\[
\begin{bmatrix}
\cos \alpha \\
\sin \alpha \\
0
\end{bmatrix}, \quad
\begin{bmatrix}
-\sin \alpha \\
\cos \alpha \\
0
\end{bmatrix}, \quad
\begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}
\]

The rotation matrix is thus:

\[
R_z(\alpha) = \begin{bmatrix}
\cos \alpha & -\sin \alpha & 0 \\
\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Similarly for the rotations around the other axes.
Consider now a point $P$ whose coordinates are expressed in two reference frames:

The coordinates of the same point in the two frames are:

$$
\begin{bmatrix}
p_x \\
p_y \\
p_z
\end{bmatrix}, \quad
\begin{bmatrix}
p'_x \\
p'_y \\
p'_z
\end{bmatrix}
$$

Therefore:

$$
\mathbf{p} = p'_x \mathbf{x}' + p'_y \mathbf{y}' + p'_z \mathbf{z}' = [\mathbf{x}' \quad \mathbf{y}' \quad \mathbf{z}'] \mathbf{p}' = R \mathbf{p}'
$$

The rotation matrix thus contains the transformation which maps the coordinates expressed in the frame $(\mathbf{x}', \mathbf{y}', \mathbf{z}')$ into the coordinates expressed in frame $(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

Inverse transformation:

$$
\mathbf{p}' = R^T \mathbf{p}
$$
Composition of rotation matrices

Let us consider three frames (denoted with 0, 1 and 2) with a common origin. We denote with:

\[ R^i_j \]  
the rotation matrix of frame \( i \) with respect to frame \( j \)

Thus:

\[ R^i_j = (R^j_i)^{-1} = (R^j_i)^T \]

The coordinates of the same point in the three frames can be expressed in different ways:

\[ p^1 = R^1_2 p^2 \quad p^0 = R^0_1 p^1 \quad p^0 = R^0_2 p^2 \]

Rotations can be obtained by composing partial rotations.

Partial rotation matrices are multiplied from left to right.
A rotation matrix represents the orientation of a frame with respect to another one by means of 9 parameters, among which 6 constraints exist.

In a **minimal representation** the orientation is described by means of 3 independent parameters.

Possible representations are:

- Euler angles (3 parameters)
- roll-pitch-yaw angles (3 parameters)
- axis/angle (4 parameters)
- quaternions (4 parameters)
With ZYZ Euler angles the sequence is composed as:

I) Rotation around Z (angle $\phi$)

II) Rotation around $Y'$ (angle $\theta$)

III) Rotation around $Z''$ (angle $\psi$)

$$R(\phi) = R_z(\phi)R_y(\theta)R_z(\psi) = \begin{bmatrix} C_\phi C_\theta C_\psi - S_\phi S_\psi & -C_\phi C_\theta S_\psi - S_\phi C_\psi & C_\phi S_\theta \\ S_\phi C_\theta C_\psi + C_\phi S_\psi & -S_\phi C_\theta S_\psi + C_\phi C_\psi & S_\phi S_\theta \\ -S_\theta C_\psi & S_\theta S_\psi & C_\theta \end{bmatrix}$$
Homogeneous representation

How can we express coordinates of point P in frame 0, based on its coordinates in frame 1?

\[ \mathbf{p}^0 = \mathbf{o}_1^0 + \mathbf{R}_1^0 \mathbf{p}^1 \]

*Rotation matrix of frame 1 w.r.t. frame 0*

Inverse transform:

\[ \mathbf{p}^1 = -\mathbf{R}_0^1 \mathbf{o}_1^0 + \mathbf{R}_0^1 \mathbf{p}^0 \]

In order to represent in a **compact form** these transformations, it is advisable to introduce a 4-dim vector:

\[ \tilde{\mathbf{p}} = \begin{bmatrix} \mathbf{w} \mathbf{p} \\ \mathbf{w} \end{bmatrix} \]

**Homogeneous representation**

\( \mathbf{w} \) is a scale factor which is always set to 1 in robotics (it is used in computer graphics).
Homogeneous transformations

Let us introduce the homogeneous transformation matrix (size $4 \times 4$):

$$ A_1^0 = \begin{bmatrix} R_1^0 & o_1^0 \\ 0^T & 1 \end{bmatrix} $$

The relationship:

$$ \vec{p}^0 = o_1^0 + R_1^0 \vec{p}^1 $$

can be expressed, in terms of homogeneous coordinates, as:

$$ \vec{p}^0 = A_1^0 \vec{p}^1 $$

$A_1^0$ relates the description (position/orientation) of a point on frame 1 with the description in frame 0.

The inverse transformation is:

$$ \vec{p}^1 = A_0^1 \vec{p}^0 = (A_1^0)^{-1} \vec{p}^0 $$

$$ A_0^1 = \begin{bmatrix} R_0^1 & -R_0^1 o_1^0 \\ 0^T & 1 \end{bmatrix} $$

N.B. $A$ is not orthogonal

Composing several transformations:

$$ \vec{p}^0 = A_1^0 A_2^1 \ldots A_n^{n-1} \vec{p}^n $$
Suppose now that rotation of one frame with respect to the second one changes with time. Let us consider a point $P$ attached to the rotating frame and expressed with the constant vector $p'$. The coordinates of the same point in the stationary frame are:

$$p(t) = R(t)p'$$

Take now the derivative with respect to time:

$$\dot{p}(t) = \dot{R}(t)p'$$

How can we express the derivative of a rotation matrix?
Derivative of a rotation matrix

Since the rotation matrix is orthogonal we have:

\[ R(t)R^T(t) = I \quad \Rightarrow \quad \dot{R}(t)R^T(t) + R(t)\dot{R}^T(t) = 0 \]

If we define the new matrix:

\[ S(t) = \dot{R}(t)R^T(t) \]

It turns out that: \( S(t) + S^T(t) = 0 \) which means that matrix \( S \) is skew symmetric.

Matrix \( S \) then takes the following form:

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

We conclude that the derivative of a rotation matrix is given by:

\[ \dot{R}(t) = S(\omega(t))R(t) \]
The derivative of vector \( p(t) \) can thus be expressed as:

\[
\dot{p}(t) = R(t)\dot{p}' = S(\omega(t))R(t)p' = S(\omega(t))p(t)
\]

On the other hand the same vector denotes the velocity of point \( P \) in the stationary frame:

\[
\dot{p}(t) = \omega(t) \times R(t)p' = \omega(t) \times p(t)
\]

- \( \omega \) is the **angular velocity** vector of the rotating frame
- symbol \( \times \) denotes **cross product**

Thus the skew symmetric matrix \( S \) can be interpreted as the **operator** that computes the cross product.
How does all this relate to the robot?

BASE

JOINTS

END EFFECTOR
The joints

Each joint allows for one (and only one) *degree of freedom* between two links. We call joint variable the coordinate associated to such degree of freedom, and then we introduce the vector of joint variables:

\[ q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \]

Schematic draws of the joints:

![Schematic draws of the joints](image)

**ROTATIONAL JOINTS**

**PRISMATIC JOINTS**
Let us define a frame attached to the base and a frame attached to the tool.

The tool frame is defined by means of three unit vectors:

- $a_e$ (approach): approach direction towards the work-piece
- $s_e$ (sliding): orthogonal to $a_e$ in the sliding plane of the gripper
- $n_e$ (normal): orthogonal to both the other ones

$p_e$ points to the origin of the tool frame (central point of the gripper).
The direct kinematic equation gives position and orientation of the tool frame w.r.t. the base frame, as a function of the joint variables.

$$T_e^b(q) = \begin{bmatrix} n_e^b(q) & s_e^b(q) & a_e^b(q) & p_e^b(q) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(homogeneous transformation matrix)

Example: planar two-link manipulator

$$T_e^b(q) = \begin{bmatrix} 0 & s_{12} & c_{12} & a_1c_1 + a_2c_{12} \\ 0 & -c_{12} & s_{12} & a_1s_1 + a_2s_{12} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
Direct kinematics

To proceed in a systematic way in the computation of the direct kinematics, a frame should be attached to each link:

\[
T_0^n(q) = A_1^0(q_1)A_2^1(q_2)\ldots A_{n-1}^{n-1}(q_{n-1})
\]

Every joint allows one degree of freedom

Proceeding iteratively:

\[
T_e^b(q) = T_0^bT_0^n(q)T_e^n
\]
Denavit-Hartenberg convention

It is a convention for the selection of the frames attached to each link.

- $z_i$ lies along the axis of joint $i+1$
- $O_i$ is at the intersection of $z_i$ axis with the common normal to axes $z_i$ e $z_{i-1}$; we denote with $O_i'$ the intersection of this common normal with axis $z_{i-1}$
- $x_i$ is aligned with the common normal to axes $z_i$ e $z_{i-1}$, with positive orientation from joint $i$ to joint $i+1$
- $y_i$ completes a right-handed frame
In order to define a frame w.r.t. to the preceding one, 4 parameters are needed.

- \(a_i\) distance of \(O_i\) from \(O_i'\)
- \(d_i\) coordinate on \(z_{i-1}\) of \(O_i'\)
- \(\alpha_i\) angle around axis \(x_i\) between axis \(z_{i-1}\) and axis \(z_i\) computed as positive counter clockwise
- \(\vartheta_i\) angle around axis \(z_{i-1}\) between axis \(x_{i-1}\) and axis \(x_i\) computed as positive counter clockwise

\(a_i\) and \(\alpha_i\) are always constant, either \(\vartheta_i\) or \(d_i\) is varying.
Denavit-Hartenberg method illustrated

https://www.youtube.com/watch?v=rA9tm0gTln8
Homogeneous transformation matrix

How to construct the transformation matrix from frame $i-1$ to frame $i$:

I) In order to superimpose frame $i-1$ to frame $i'$ we translate the frame along axis $z_{i-1}$ by a length $d_i$ rotating by an angle $\theta_i$ around $z_{i-1}$:

$$
A_{i-1}^{i'} = \begin{bmatrix}
c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\
s_{\theta_i} & c_{\theta_i} & 0 & 0 \\
0 & 0 & 1 & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

II) In order to superimpose frame $i'$ to frame $i$ we translate the frame along axis $x_i'$ by a length $a_i$, rotating of an angle $\alpha_i$ around $x_i'$:

$$
A_i^{i'} = \begin{bmatrix}
1 & 0 & 0 & a_i \\
0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\
0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

$$
A_{i-1}(q_i) = A_{i-1}^{i'} A_i^{i'} = \begin{bmatrix}
c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\
s_{\theta_i} c_{\alpha_i} & c_{\theta_i} & -s_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\
0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}
$$
Joint space and operational space

The **joint space** is defined by the vector of joint variables:

\[
q = \begin{bmatrix}
q_1 \\
\vdots \\
q_n
\end{bmatrix}
\]

\[q_i = \theta_i\text{ (rotating joint)}\]

\[q_i = d_i\text{ (prismatic joint)}\]

The **operational space** is the space where the task that the manipulator has to accomplish is specified. It is defined by the posture \(x\):

\[
x = \begin{bmatrix}
p \\
\phi
\end{bmatrix}
\]

\[p\text{ (position)}\]

\[\phi\text{ (minimal representation of the orientation)}\]

\[m\text{ components}\]

Direct kinematic relation: \(x = k(q)\)
Three d.o.f. planar manipulator

We can define the orientation with the angle $\phi$ formed by the end effector (vector $x_3$) with axis $x_0$

$$T_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix}
c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\
s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$x = \begin{bmatrix}
p_x \\
p_y \\
\phi
\end{bmatrix} = k(q) = \begin{bmatrix}
a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\
a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\
\theta_1 + \theta_2 + \theta_3
\end{bmatrix}$$
A six d.o.f. robot
Inverse kinematics problem

\[ T \Rightarrow q \quad \text{Given position and orientation of the tool frame, find} \]
\[ x \Rightarrow q \quad \text{the corresponding joint variables.} \]

- The problem may admit no solutions (if position and orientation do not belong to the workspace of the manipulator)
- The analytical solution (in closed form) may not exist. In this case numerical techniques are used
- Multiple or an infinite number of solutions might exist

In general the solution is found without a systematic procedure, rather relying on intuition in manipulating the equations.
Two d.o.f. planar manipulator

\[ p_x = a_1 \cos(\vartheta_1) + a_2 \cos(\vartheta_1 + \vartheta_2) \]
\[ p_y = a_1 \sin(\vartheta_1) + a_2 \sin(\vartheta_1 + \vartheta_2) \]

Squaring and summing:
\[ c_2 = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \quad \Rightarrow \quad \vartheta_2 = \text{Atan} \, 2(s_2, c_2) \]
\[ s_2 = \pm \sqrt{1 - c_2^2} \quad \text{2 solutions} \]

\[ c_1 = \frac{(a_1 + a_2 c_2) p_x + a_2 s_2 p_y}{p_x^2 + p_y^2} \]
\[ s_1 = \frac{(a_1 + a_2 c_2) p_y - a_2 s_2 p_x}{p_x^2 + p_y^2} \quad \Rightarrow \quad \vartheta_1 = \text{Atan} \, 2(s_1, c_1) \]

\text{COMPLICATED!}
Anthropomorphic manipulator

- Eight admissible configurations exist
- Right/left shoulder
- Up/down elbow
- Up/down wrist
Differential kinematics: geometrical Jacobian

Let’s introduce now the linear velocity and the angular velocity of the tool frame (attached to the tool): \( \dot{p} \) and \( \omega \).

The goal of **differential kinematics** is to express these velocities in terms of the joint velocities.

\[
\dot{p} = J_P(q) \dot{q} \\
\omega = J_O(q) \dot{q}
\]

In a compact form: 

\[
v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q) \dot{q}
\]

The \((6 \times n)\) matrix:

\[
J(q) = \begin{bmatrix} J_P(q) \\ J_O(q) \end{bmatrix}
\]

is called **geometrical Jacobian** of the manipulator. Systematic methods exist to compute the Jacobian.
Analytical Jacobian

Let’s go back to the direct kinematic equation of a manipulator:

\[ x = k(q) = \begin{bmatrix} p(q) \\ \phi(q) \end{bmatrix} \]

where \( \phi \) is a minimal representation of the orientation. Differentiating w.r.t. time we obtain:

\[ \dot{x} = \frac{\partial k(q)}{\partial q} \dot{q} = J_A(q) \dot{q} \]

On the other hand:

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

Matrix: \( J_A(q) = \begin{bmatrix} J_P(q) \\ J_\phi(q) \end{bmatrix} \) is called analytical Jacobian of the manipulator.
Analytical vs. geometrical Jacobian

The link between the angular velocity $\omega$ and the derivative of vector $\phi$ expressing the orientation is the following one:

$$\omega = T(\phi)\dot{\phi}$$

where $T$ is a matrix that depends on the representation of the orientation:

$$T(\phi) = \begin{bmatrix} 0 & -s_{\phi} & c_{\phi}s_{\theta} \\ 0 & c_{\phi} & s_{\phi}s_{\theta} \\ 1 & 0 & c_{\theta} \end{bmatrix} \quad \text{(for the ZYZ Euler angles)}$$

Let us thus express the velocity (linear and angular) of the tool frame in terms of the derivatives of $p$ and $\phi$:

$$\mathbf{v} = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & T(\phi) \end{bmatrix} \dot{\mathbf{x}} = T_A(\phi)\dot{\mathbf{x}} = T_A(\phi)J_A\dot{\mathbf{q}}$$

The relation between analytical and geometrical Jacobian follows: $\mathbf{J} = T_A(\phi)J_A$
Kinematic singularities

The equation defining the geometrical Jacobian is:

\[ \mathbf{v} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \]

The values of \( q \) for which matrix \( \mathbf{J} \) is rank-deficient are called **kinematic singularities**. At a kinematic singularity we have:

1. Loss of mobility (it is not possible to impose arbitrary motion laws)
2. Possibility of infinite solutions to the kinematic inversion problem
3. High velocities in joint space (around the singularity)

The singularities may happen:

1. **At the borders** of the manipulator work-space
2. **Inside** the manipulator work-space

The latter are more problematic, since they can be incurred with trajectories planned in the operational space.
Kinematic singularities: example

For a two-link manipulator the Jacobian is:

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

We can compute singularities:

\[ \det(J) = a_1 a_2 s_2 = 0 \iff \theta_2 = \begin{cases} 0 \\ \pi \end{cases} \]

These are singularities at the borders of the workspace.

In these configurations the two columns of the Jacobian are not independent.
Kinematic singularities of a complete manipulator

Arm singularity

Singularity at the intersection of the wrist center and axis 1

Rotation center of axis 1

Source: ABB

Elbow singularity

Axis 5 with an angle of 0 degrees

Axis 6 parallel to axis 4

Wrist singularity
Consequences of kinematic singularities on robot motion

https://www.youtube.com/watch?v=zIGCurgsqg8
Inversion of the differential kinematics

The differential kinematics is linear for a certain value of $q$:

$$v = J(q)\dot{q}$$

Given a velocity $v$ in the operational space and an initial condition on $q$ we might solve the kinematic inversion problem by inverting the differential kinematics and then integrating. If the Jacobian is square ($n = r$, number of coordinates in the operational space needed to describe a task):

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

However, using this expression directly, drifts of the solution may occur. The error in the operational space made by the kinematic inversion algorithm is then introduced:

$$e = x_d - x$$
If we adopt the following dependence of $\dot{q}$ from $e$:

$$\dot{q} = J_A^{-1}(q)(\dot{x}_d + Ke)$$

we obtain:

$$\dot{e} + Ke = 0$$

and the diagram:

Mathematically, this corresponds to solve the inverse kinematics problem through a Gauss-Newton iterative method. Proof of convergence is trivial as:

$$\dot{e} + Ke = 0$$
If we adopt the following (simpler) dependence:

$$\dot{q} = J_A^T(q)Ke$$

we obtain the diagram:

Mathematically, this corresponds to solve the inverse kinematics problem through a gradient descent iterative method. Proof of convergence can be obtained through a Lyapunov argument.