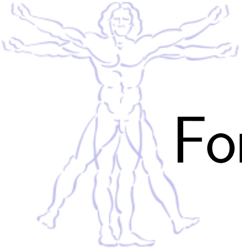


Forward kinematics

From joints to kinematics

Last update: April 26, 2023

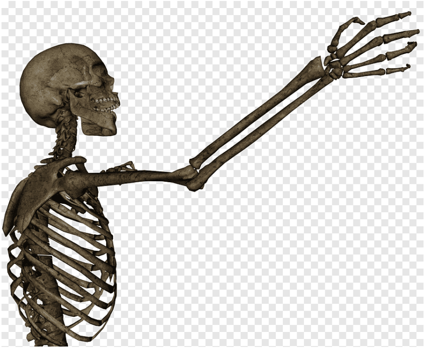
Agenda

- Quantifying human motion
- Coordinate frames
- Transformation matrices
- From frame to frame
- Forward kinematics model

Human motion

Pose

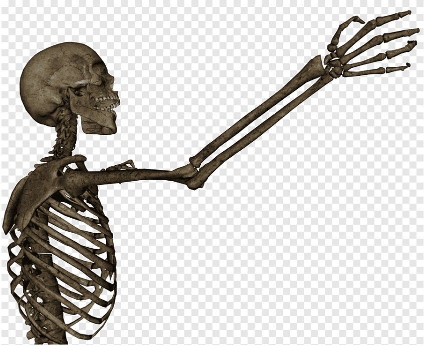
Description of position and orientation of segments, with respect to a reference frame



Human motion

Pose

Description of position and orientation of segments, with respect to a reference frame



We use coordinate frames

Coordinate systems

Cartesian coordinates

In simple words

A coordinate system is a mathematical tool that allows us to describe the position of objects in space using numbers. Each coordinate system has axes, equal in number to the number of dimensions of space.

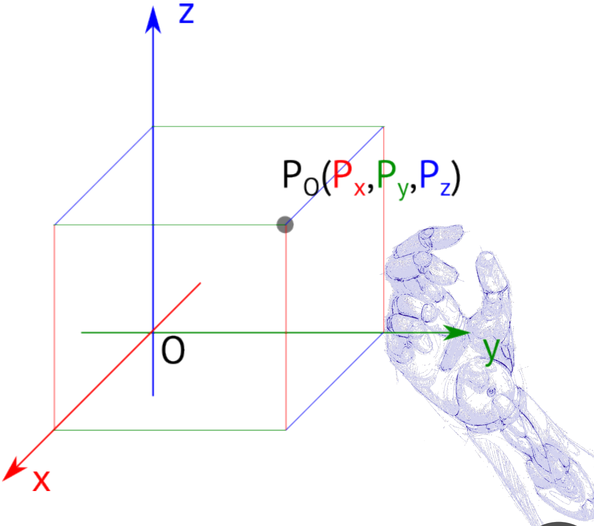
Properties

- The axes must be perpendicular to each other
- The length of the axes is one unit
- Each point has n number of coordinates, equal to the number of axes
- There can be more than one coordinate system to describe a certain space

Coordinate systems

The $\mathbb{R}^3$ case

In three dimensional space (3D), we need three axes to describe the position of each point. Each of these axes must be perpendicular to the other two.



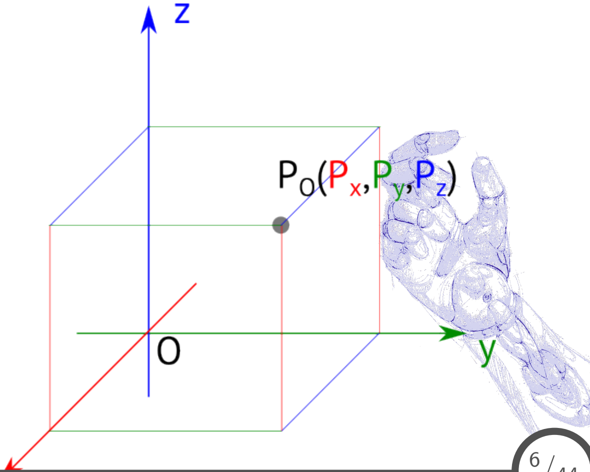
Points

Description of points

Since we might have different coordinate frames defined, we need to define the notation to describe the potision of a point P in respect to a coordinate frame

For a point P described in coordinate frame O, we will use the following notation to describe its position

$$P_O = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$



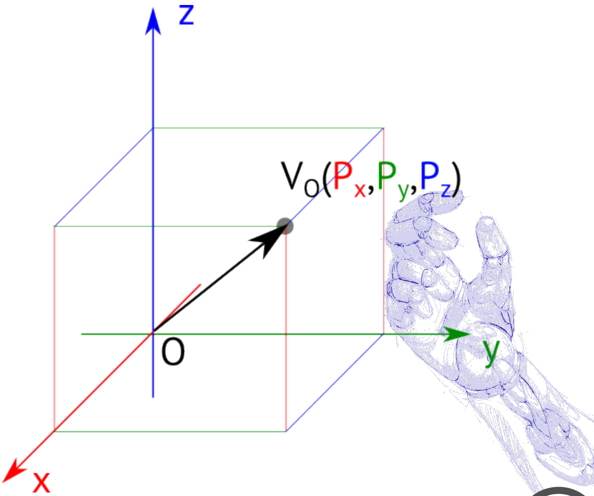
Vectors

Description of vectors

Vectors are just like points!

A vector V described in coordinate frame O, is totally defined by its end point P and we use the same notation as points

$$V_O = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

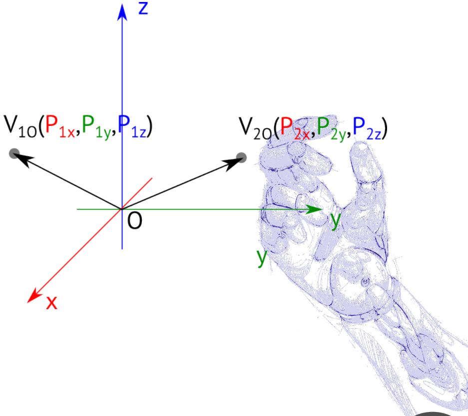


Vectors

Description of vectors

When we have multiple vectors, we can group them together

$$V_O = \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} P_1x & P_2x \\ P_1y & P_2y \\ P_1z & P_2z \end{bmatrix}$$

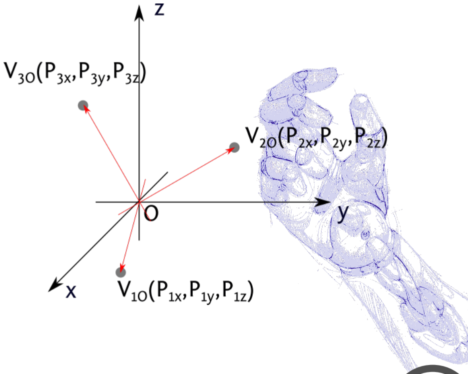


Coordinate frames

Description of coordinate frames

A coordinate system (a.k.a coordinate frame) is a set of three vectors. Therefore, we can describe it in respect to another coordinate frame using the notation we know

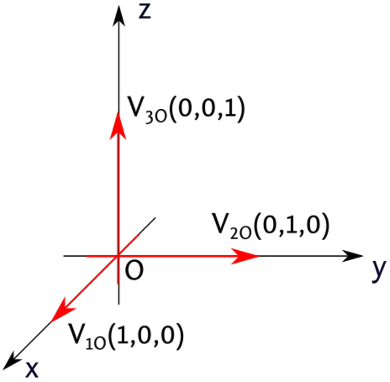
$$V_O = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix} = \begin{bmatrix} P_1x & P_2x & P_3x \\ P_1y & P_2y & P_3y \\ P_1z & P_2z & P_3z \end{bmatrix}$$



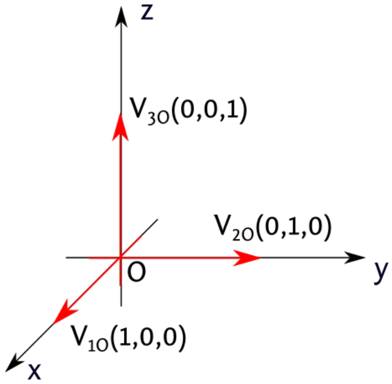
Coordinate frames

Description of coordinate frames

In the special case, when the axes of the two coordinate frames are aligned, we end up with....



In the special case, when the axes of the two coordinate frames are aligned, we end up with....



$$V_O = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The identity matrix!

Transformations

A nice trick to move things around

Since we talk about motion, we need to define a way to move things around. To do this, we use matrices.

Definition of transformation matrix R for a counter-clockwise rotation θ in $\mathbb{R}^2$:

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Transformations

A nice trick to move things around

Let's put this in practice. Suppose we have a point $P_O = \begin{bmatrix} P_x \\ P_y \end{bmatrix}$, and we want to rotate it by θ degrees. All we need to do is to multiply the transformation matrix R with the point P_O. The result of the multiplication is the transformed point $P_O^{'}$

$$P_O^{'} = R * P_O$$

Transformations

Example

Suppose we have a point $P_O = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and we want to rotate it around the origin of the axes by $\theta = 90°$:

$$P'_O = R * P_O = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} P_x \\ P_y \end{bmatrix}$$

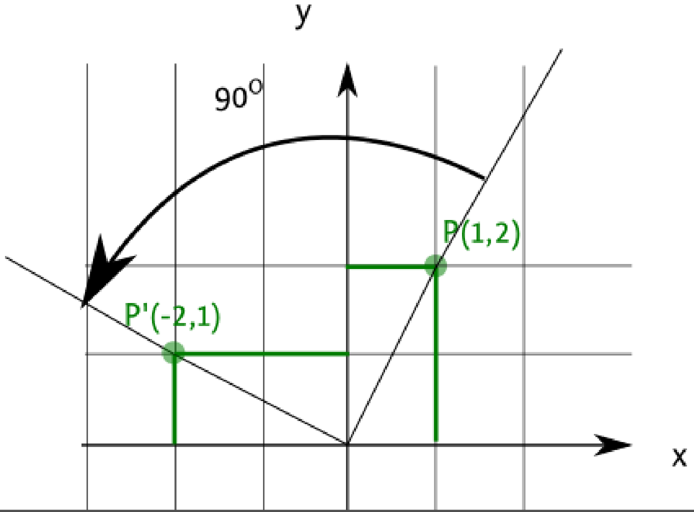
$$= \begin{bmatrix} cos90 & -sin90 \\ sin90 & cos90 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$

Transformations

Example

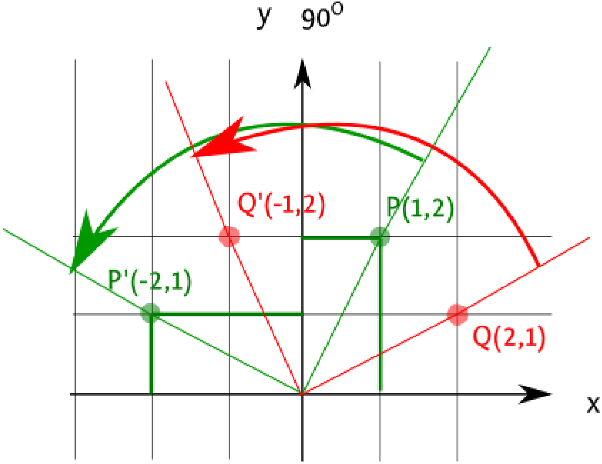
$$P'_O = R * P_O = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$



Transformations

It works with more points too!

$$P'_O = R * P_O = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 1 & 2 \end{bmatrix}$$



Transformations

Let's do it in 3D

Transformations in $\mathbb{R}^3$ follow the same logic. There are three rotations that can be applied in three dimensions, each around one of the three axes. Rotation around axis:

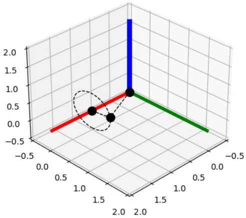
$$R(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix}$$

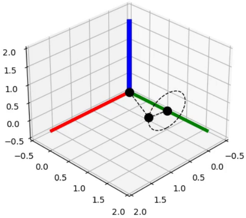
$$R(y,\phi) = \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix}$$

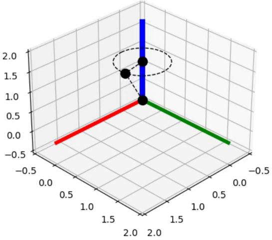
$$R(z,\omega) = \begin{bmatrix} cos\omega & -sin\omega & 0 \\ sin\omega & cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Transformations

Let's do it in 3D







Transformations

What about translation?

The second type of basic transformation is the translation. How do we 'apply' translations to a point?

Homogenious transformation matrix:

$$T = \left[\begin{array}{ccc|c} 3 & \times & 3 & 3 \times 1 \\ \hline 1 & \times & 3 & 1 \times 1 \end{array}\right] = \left[\begin{array}{ccc|c} & rotation & & \begin{array}{c} trans- \\ la- \\ tion \end{array} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \quad (1)$$

Transformations

Homogenious translations

$$Trans(X,a) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans(Y,b) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Trans(Z,c) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformations

Homogenious rotations

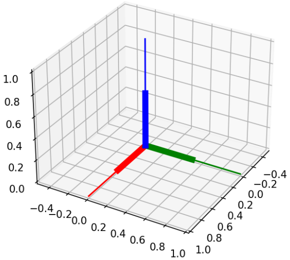
$$Rot(X,\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sin\theta & 0 \\ 0 & sin\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(Y,\phi) = \begin{bmatrix} cos\phi & 0 & sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -sin\phi & 0 & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(Z,\omega) = \begin{bmatrix} cos\omega & -sin\omega & 0 & 0 \\ sin\omega & cos\omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As we already saw, we use a matrix notation to express a coordinate frame relative to another. A coordinate frame aligned with a basis coordinate frame is expressed with the identity matrix.

$$V_O = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

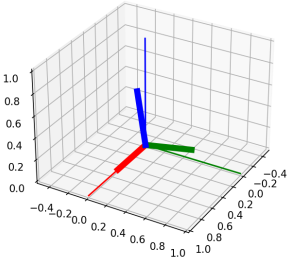


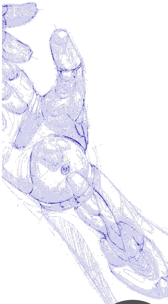
Transformations

Transforming Coordinate Frames

We can transform the coordinate frame by multiplying its matrix representation with transformation matrices corresponding to the transformation we want.

$$V_O' = Rot(X, \theta) * V_O$$



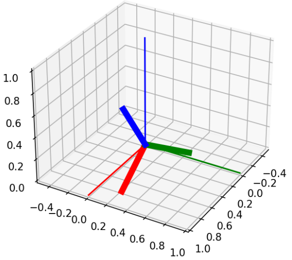


Transformations

Transforming Coordinate Frames

We can apply multiple transformations by multiplying the resulting coordinate frame with a second transformation matrix.

$$V_O' = Rot(Y,\phi) * Rot(X,\theta) * V_O$$

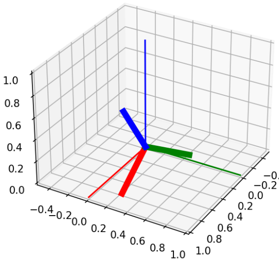


Transformations

Transforming Coordinate Frames

We can apply multiple transformations by multiplying the resulting coordinate frame with a second transformation matrix.

$$V_O' = Rot(Y, \phi) * Rot(X, \theta) * V_O$$

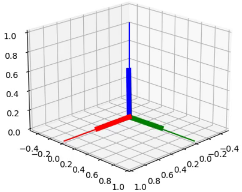


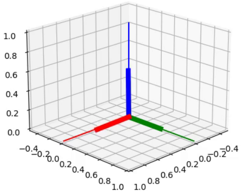
Arbitrary position

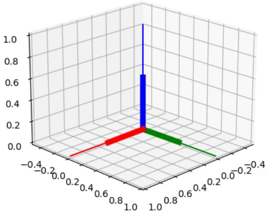
The transition between two arbitrarily positioned coordinate frames can **always** be described in terms of elementary transformations (rotations and translations)

Transformations

Transforming Coordinate Frames





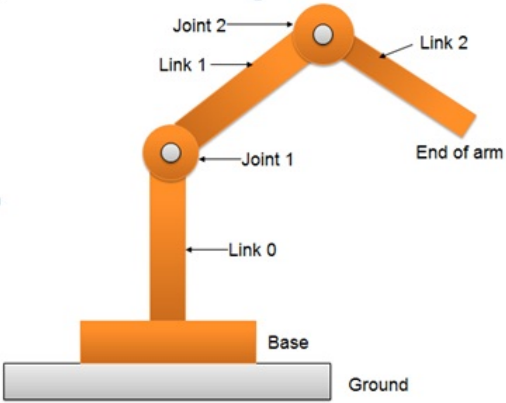


Arbitrary position

The transition between two arbitrarily positioned coordinate frames can **always** be described in terms of elementary transformations (rotations and translations)

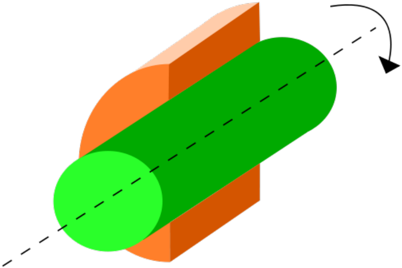
Links

A link is a rigid part of a mechanism. In our case, we consider bones as links



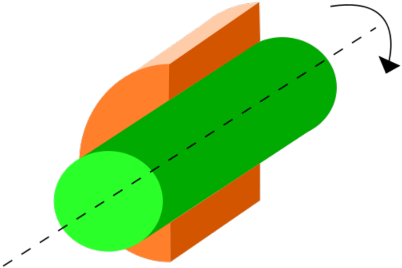
Joints

A revolute joint is a joint that allows motion that changes the orientation of a segment by rotating around a fixed axis. We usually work with one degrees of freedom joints (e.g. hinge).



Joints

A revolute joint is a joint that allows motion that changes the orientation of a segment by rotating around a fixed axis. We usually work with one degrees of freedom joints (e.g. hinge).



If we have a more complex joint (i.e. spherical), then we model it as subsequent hinge joints.

Forward kinematics

Definition

The Forward kinematics (FK) is a mathematical tool that allows us to calculate the position and orientation (pose) of a body's point of interest if we know the state of the joints and the lengths of the links.

Forward kinematics

Definition

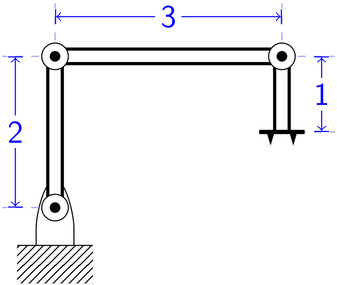
The Forward kinematics (FK) is a mathematical tool that allows us to calculate the position and orientation (pose) of a body's point of interest if we know the state of the joints and the lengths of the links.

In simple words

How do I calculate the pose of the human arm if I know the joint angles?

Forward kinematics

Definition



How do we calculate the pose of the end-effector of this arm?

Forward kinematics

Definition

We describe the pose of the end-effector using a 4x4 transformation matrix (contains information about position and orientation).

$$T = \left[\begin{array}{ccc|c} 3 & \times & 3 & 3\times 1 \\ \hline 1 & \times & 3 & 1\times 1 \end{array}\right] = \left[\begin{array}{ccc|c} & rotation & & \begin{matrix} trans- \\ la- \\ tion \end{matrix} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{2}$$

Forward kinematics

Calculation

To define the FK we perform the following steps:

Forward kinematics

Calculation

To define the FK we perform the following steps:

- We identify the links and joints of the arm.

Forward kinematics

Calculation

To define the FK we perform the following steps:

- We identify the links and joints of the arm.
- We attach a fixed coordinate frame in a convenient location.

Forward kinematics

Calculation

To define the FK we perform the following steps:

- We identify the links and joints of the arm.
- We attach a fixed coordinate frame in a convenient location.
- We attach a coordinate frame on each link at their joints.

Forward kinematics

Calculation

To define the FK we perform the following steps:

- We identify the links and joints of the arm.
- We attach a fixed coordinate frame in a convenient location.
- We attach a coordinate frame on each link at their joints.
- We calculate the transformation between each subsequent coordinate frame.

Forward kinematics

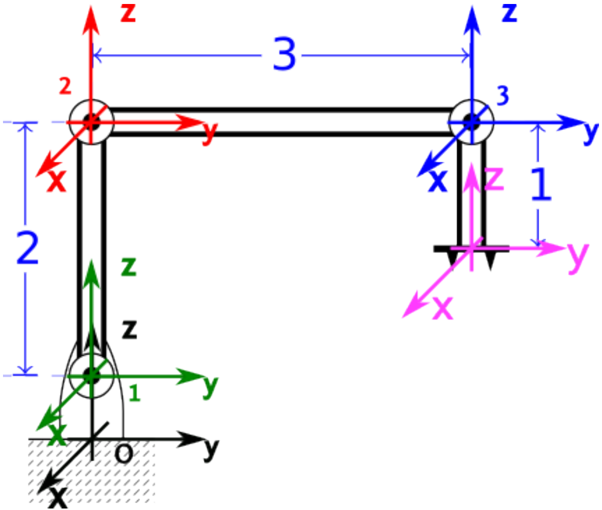
Calculation

To define the FK we perform the following steps:

- We identify the links and joints of the arm.
- We attach a fixed coordinate frame in a convenient location.
- We attach a coordinate frame on each link at their joints.
- We calculate the transformation between each subsequent coordinate frame.
- We combine the transformations to calculate the overall transformation from base to end-effector.

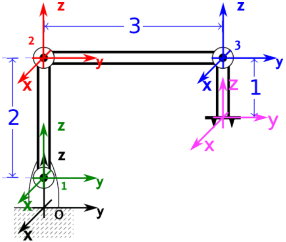
Forward kinematics

Calculation



Forward kinematics

Static calculation

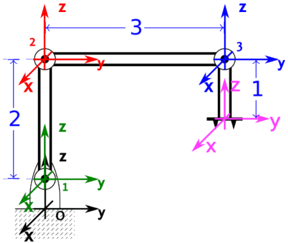


$$R_0^1 = Trans(Z, 1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_1^2 = Trans(Z, 2) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

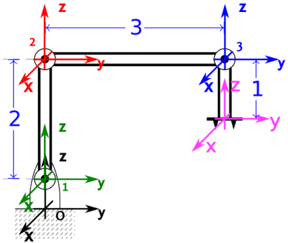
Static calculation



$$R_2^3 = Trans(Y,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \; R_3^4 =$$

Forward kinematics

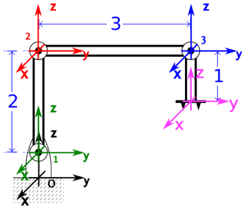
Static calculation



$$R_2^3 = Trans(Y,3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

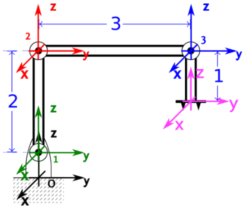
Forward kinematics

Static calculation



Forward kinematics

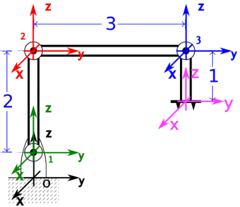
Static calculation



$$R_0^4 = R_0^1 * R_1^2 * R_2^3 * R_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

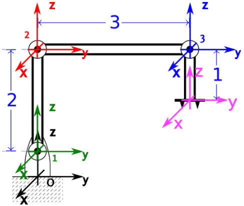
Static calculation



$$R_0^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & & & trans- \\ & rotation & & la- \\ & & & tion \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

Static calculation

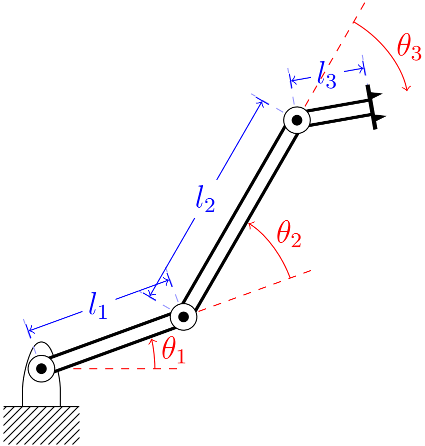


$$R_0^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & rotation & & trans- \\ & & & la- \\ & & & tion \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_m^n = \begin{bmatrix} X_x & Y_x & Z_x & P_x \\ X_y & Y_y & Z_y & P_y \\ X_z & Y_z & Z_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

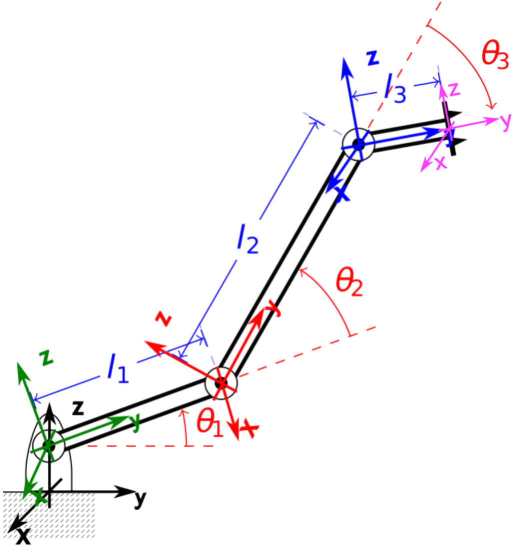
Forward kinematics

What about other configurations?



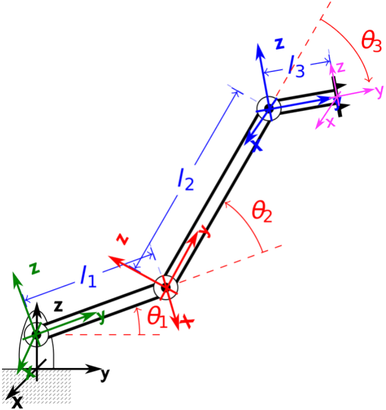
Forward kinematics

Dynamic Calculation



Forward kinematics

Dynamic calculation



$$R_0^1 = Trans(Z,1) * R(X,\theta_1) =$$

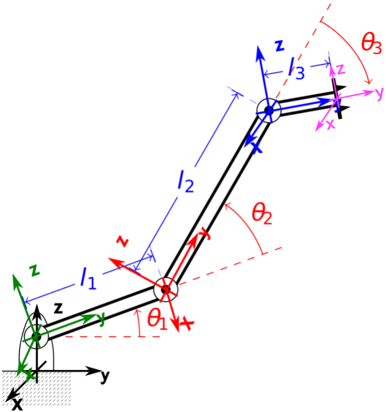
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_1 & -s_1 & 0 \\ 0 & s_1 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_1^2 = Trans(Y,l_1) * R(X,\theta_2) =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & -s_2 & 0 \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

Dynamic calculation



$$R_2^3 = Trans(Y, l_2) * R(X, \theta_3) =$$

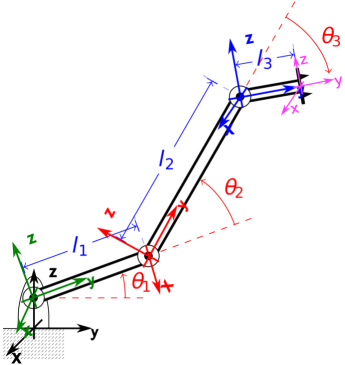
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_3 & -s_3 & 0 \\ 0 & s_3 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_0^4 = R_0^1 * R_1^2 * R_2^3 * R_3^4$$

Forward kinematics

Dynamic calculation



$$R_3^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{1,2,3} & -s_{1,2,3} & l_3 c_{1,2,3} + l_2 c_{1,2} + l_1 c_1 \\ 0 & s_{1,2,3} & c_{1,2,3} & l_3 s_{1,2,3} + l_2 s_{1,2} + l_1 s_1 + 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

Dynamic calculation

$$R_0^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{1,2,3} & -s_{1,2,3} & l_3 c_{1,2,3} + l_2 c_{1,2} + l_1 c_1 \\ 0 & s_{1,2,3} & c_{1,2,3} & l_3 s_{1,2,3} + l_2 s_{1,2} + l_1 s_1 + 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward kinematics

The FK is a transformation matrix, a function of the joint positions and link lengths. If we know these variables, we can calculate the position and orientation of the end effector (or any other point).

Forward kinematics

Spherical joints

How do we calculate the forward kinematics of a spherical joint?

Questions?