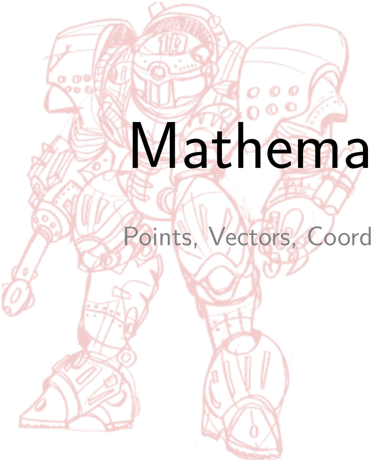


Mathematical background

Points, Vectors, Coordinate systems, Transformation matrices

TECHNICAL
UNIVERSITY
OF CLUJ-NAPOCA
ROMANIA

Last update: October 5, 2023

Tassos Natsakis tassos.natsakis@aut.utcluj.ro

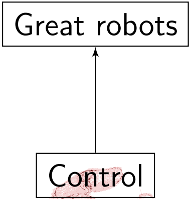
Grand scheme

The big picture

Great robots

Grand scheme

The big picture

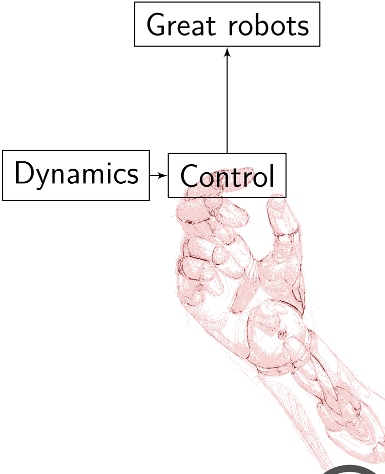


Great robots

Control

Grand scheme

The big picture



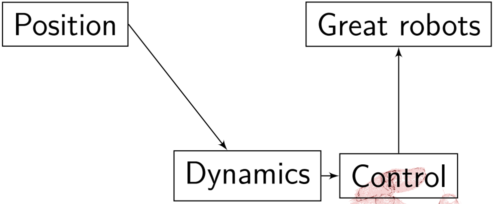
Great robots

Dynamics

Control

Grand scheme

The big picture



Position

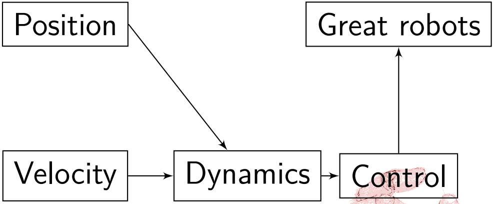
Dynamics

Control

Great robots

Grand scheme

The big picture



Position

Great robots

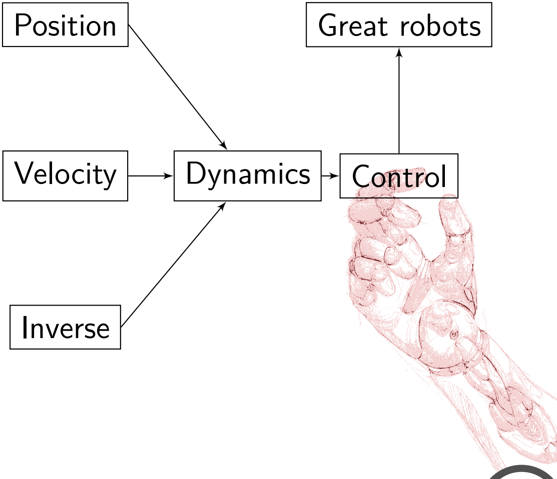
Velocity

Dynamics

Control

Grand scheme

The big picture



Position

Velocity

Inverse

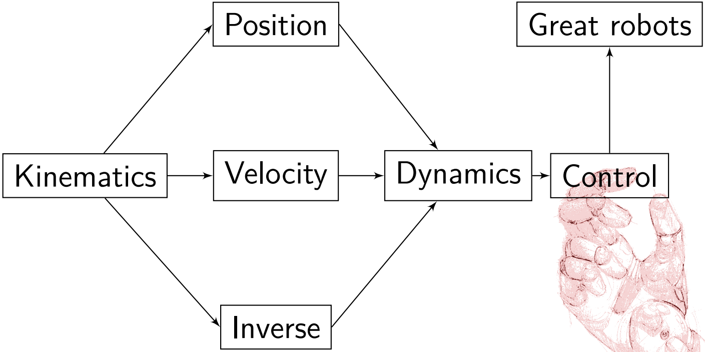
Dynamics

Control

Great robots

Grand scheme

The big picture



Position

Great robots

Kinematics

Velocity

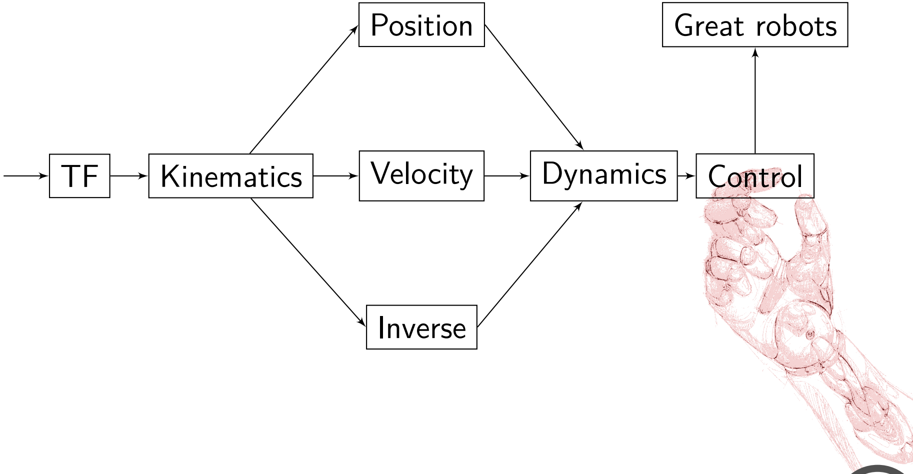
Dynamics

Control

Inverse

Grand scheme

The big picture



Agenda

- Definitions of coordinate systems
- Rule of the right hand
- Points and vectors in $\mathbb{R}^2$ & $\mathbb{R}^3$
- Transformation matrices
- Homogeneous transformations

Coordinate systems

Cartesian coordinates

In simple words

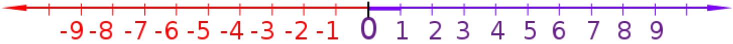
A coordinate system is a mathematical tool that allows us to describe the position of objects in space using numbers. Each coordinate system has axes, equal in number to the number of dimensions of space.

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 $\mathbb{R}^1$ case



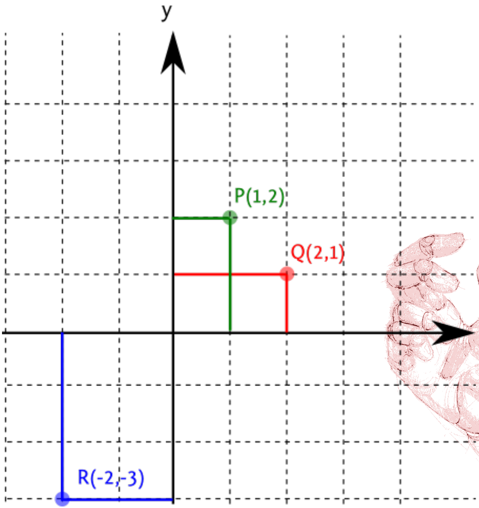
A point P in $\mathbb{R}^1$ space is represented as $P = [2]$

Coordinate systems

The $\mathbb{R}^2$ case

A point P in $\mathbb{R}^2$ space is represented as

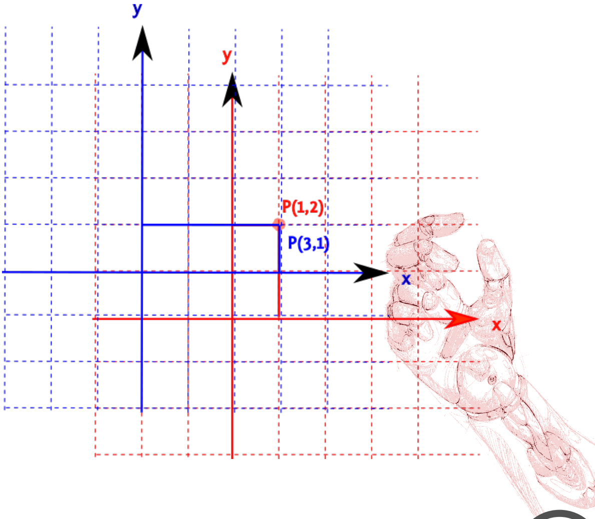
$$P = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$



Coordinate systems

The $\mathbb{R}^2$ case

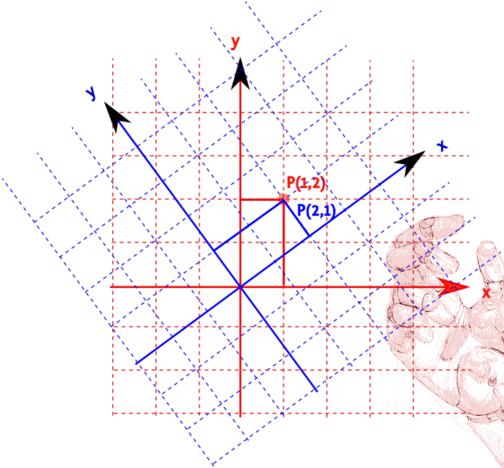
We can have different coordinate systems describing the same points. The coordinate systems might be translated relative to each other



Coordinate systems

The $\mathbb{R}^2$ case

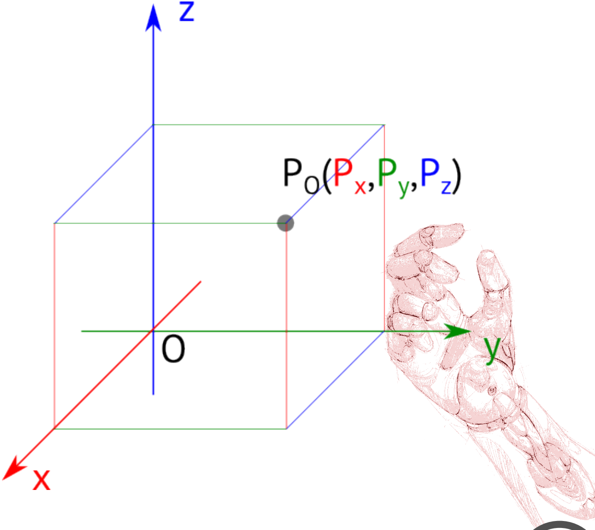
Or they can be rotated relative to each other



Coordinate systems

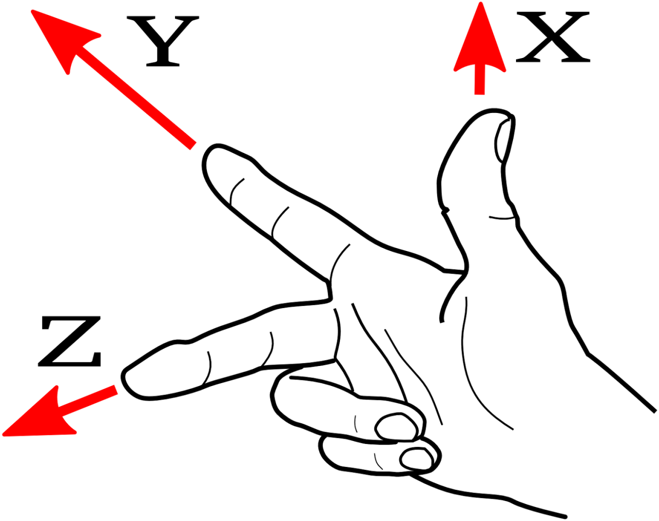
The $\mathbb{R}^3$ case

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.



Coordinate systems

The rule of the right hand



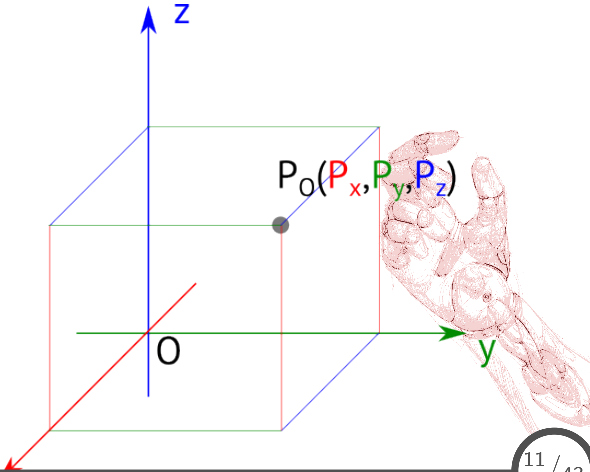
Points

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

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



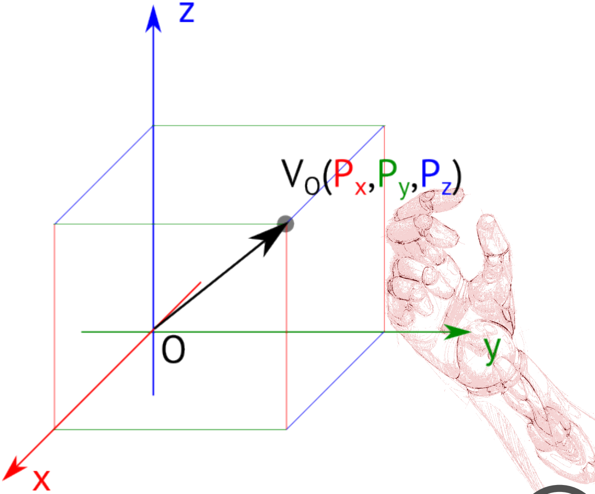
Vectors

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

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

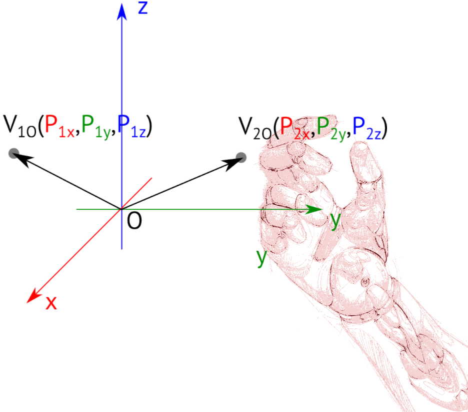


Vectors

Description of vectors

When we have multiple vectors, we can group them together

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

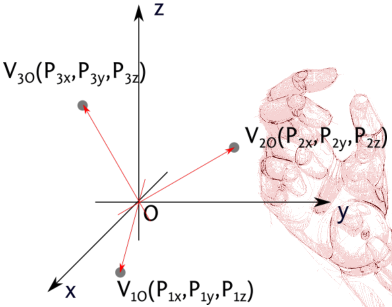


Coordinate frames

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

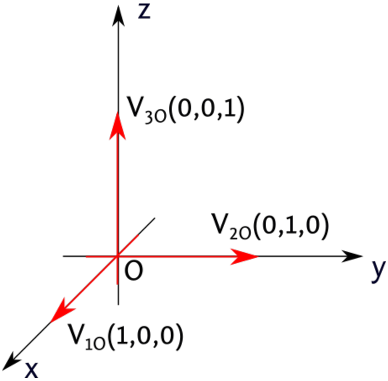
$$V_O = \begin{bmatrix} V_1 & V_2 & V_3 \end{bmatrix} = \begin{bmatrix} P_1x & P_2x & P_3x \\ P_1y & P_2y & P_3y \\ P_1z & P_2z & P_3z \end{bmatrix}$$


z
$V_{3O}(P_{3x},P_{3y},P_{3z})$
$V_{2O}(P_{2x},P_{2y},P_{2z})$
O
y
x
$V_{1O}(P_{1x},P_{1y},P_{1z})$

Coordinate frames

Description of coordinate frames

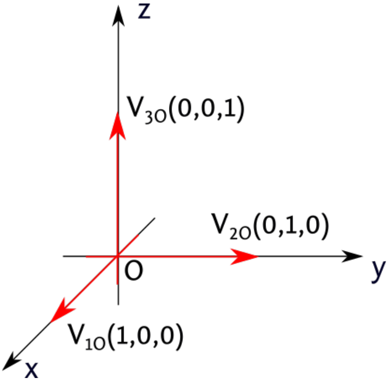
In the special case, when the axes of the two coordinate frames are aligned, we end up with....



Coordinate frames

Description of coordinate frames

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

Robots are about motion, so 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 $P_O^{'}$

$$P_O^{'} = RP_O$$

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 $\theta = 90^\circ$:

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

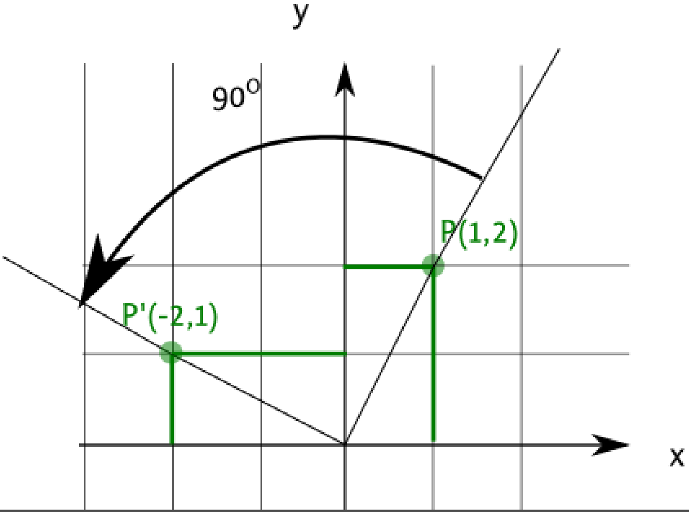
$$= \begin{bmatrix} cos90 & sin90 \\ -sin90 & cos90 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

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

Transformations

Example

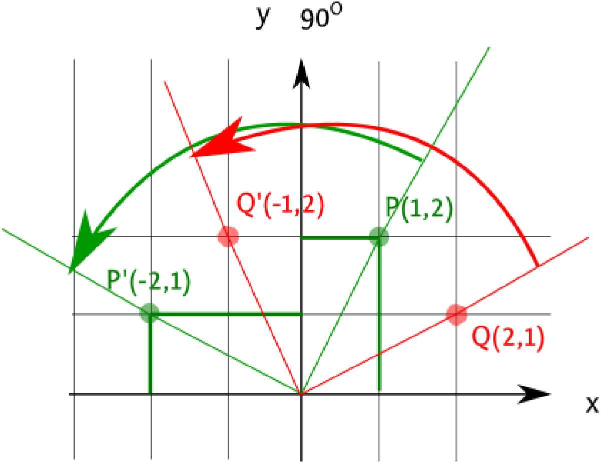
$$P'_O = RP_O = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$$



Transformations

It works with more points too!

$$PQ'_O = RPQ_O = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ 1 & 2 \end{bmatrix}$$



Transformations

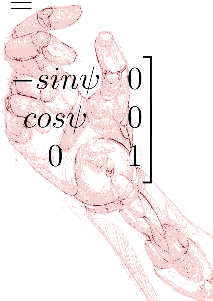
Let's do it in 3D

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:

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

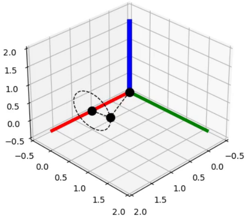
$$R(y,\phi) = \begin{bmatrix} cos\phi & 0 & sin\phi \\ 0 & 1 & 0 \\ -sin\phi & 0 & cos\phi \end{bmatrix}$$

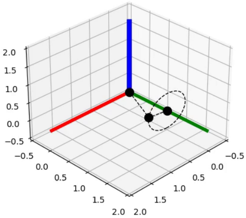
$$R(z,\psi) = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

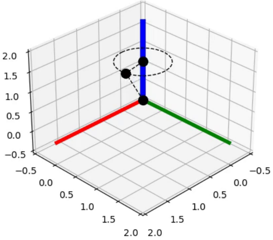


Transformations

Let's do it in 3D







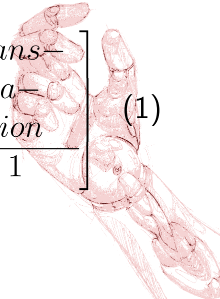
Transformations

What about translation?

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

Homogeneous transformation matrix:

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



Transformations

Homogeneous translations

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

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

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

Tassos Natsakis tassos.natsakis@aut.utcluj.ro

Transformations

Homogeneous rotations

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

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

$$RotZ(\psi) = \begin{bmatrix} cos\psi & -sin\psi & 0 & 0 \\ sin\omega & cos\omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformations

Homogeneous transformations

Since the homogeneous matrix for a $\mathbb{R}^3$ transformation is a 4x4 matrix, we need to define points and vectors as 4x1, in order for the multiplication to be possible. Therefore, points are defined as:

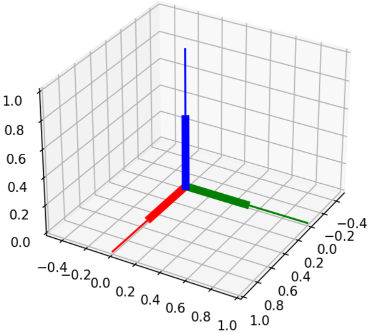
$$P_O = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$$

Transformations

Transforming Coordinate Frames

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.

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

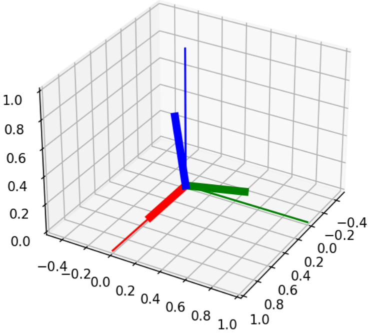


Transformations

Transforming Coordinate Frames

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

$$V_O' = RotX(\theta)V_O$$

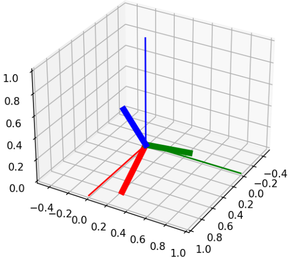


Transformations

Transforming Coordinate Frames

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

$$V_O' = RotY(\phi)RotX(\theta)V_O$$

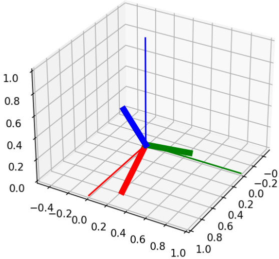


Transformations

Transforming Coordinate Frames

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

$$V_O' = RotY(\phi)RotX(\theta)V_O$$

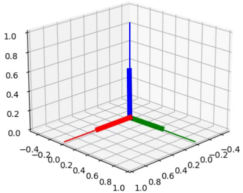


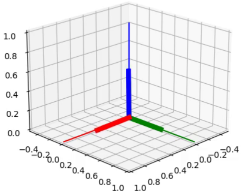
Arbitrary position

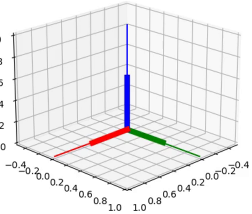
The transition between two arbitrarily positioned coordinate frames can **always** be described in terms of elementary transformations (rotations and translations)

Transformations

Transforming Coordinate Frames





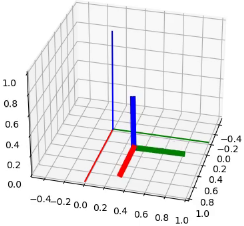


Transformations

Multiplication from left

Multiplication from the left results in transformation according to the axes of the base coordinate frame.

$$V_O' = RotX(\theta)V_O$$

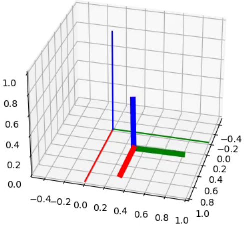


Transformations

Multiplication from right

Multiplication from the right results in transformation according to the axes of the transformed coordinate frame.

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

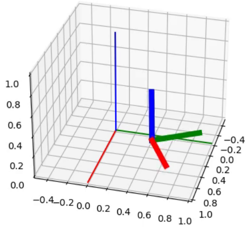


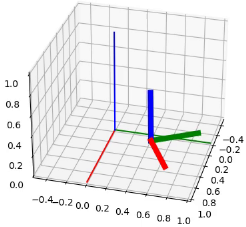
Transformations

Left and right multiplication

$$V_O' = RotX(\theta)V_O$$

$$V_O' = V_O RotX(\theta)$$



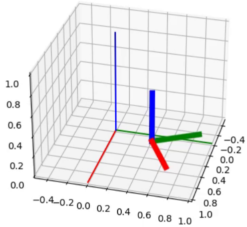


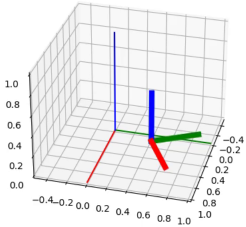
Transformations

Left and right multiplication

$$V_O' = Trans(X, d)V_O$$

$$V_O' = V_O Trans(X, d)$$



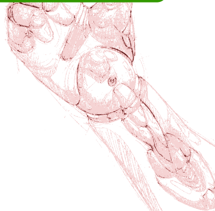


Transformations

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)



Transformations

Transforming Coordinate Frames

Arbitrary **orientation**

The transition between two arbitrarily **oriented** coordinate frames can **always** be described in terms of **three elementary rotations**

Transformations

Transforming Coordinate Frames

Arbitrary **orientation**

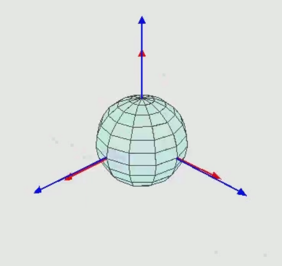
The transition between two arbitrarily **oriented** coordinate frames can **always** be described in terms of **three elementary rotations**

Transformations

Transforming Coordinate Frames

Arbitrary **orientation**

The transition between two arbitrarily **oriented** coordinate frames can **always** be described in terms of **three elementary rotations**

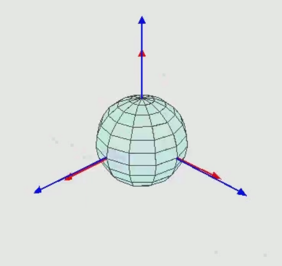


Transformations

Transforming Coordinate Frames

Arbitrary **orientation**

The transition between two arbitrarily **oriented** coordinate frames can **always** be described in terms of **three elementary rotations**



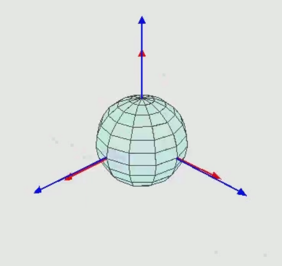
Three subsequent rotations, along specific axes

Transformations

Transforming Coordinate Frames

Arbitrary **orientation**

The transition between two arbitrarily **oriented** coordinate frames can **always** be described in terms of **three elementary rotations**



Three subsequent rotations, along specific axes

- Z-X-Z
- X-Y-X
- X-Y-Z
- ...

Transformations

Quaternions

Quaternions

Quaternions is a number system that extends the complex numbers. It can be used to represent relative orientation of two coordinate frames

$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$

Transformations

Quaternions

Quaternions

Quaternions is a number system that extends the complex numbers. It can be used to represent relative orientation of two coordinate frames

$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$

x	**1**	**i**	**j**	**k**
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Transformations

Quaternions

Quaternions

Quaternions is a number system that extends the complex numbers. It can be used to represent relative orientation of two coordinate frames

$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$

x	**1**	**i**	**j**	**k**
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Trust me, it is weird

Transformations

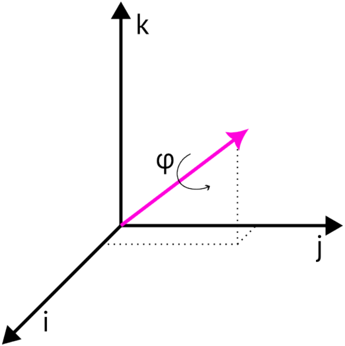
Quaternions

$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

Transformations

Quaternions

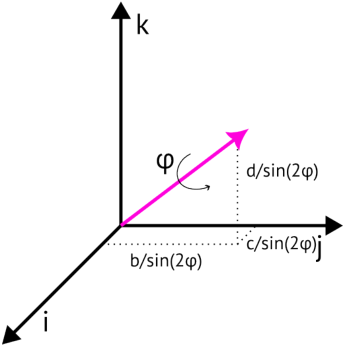
$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$



Transformations

Quaternions

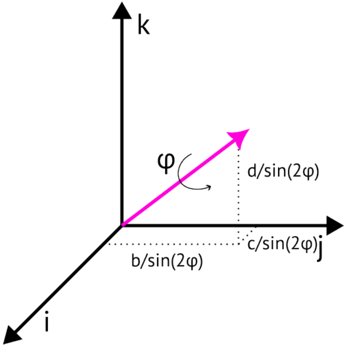
$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$



Transformations

Quaternions

$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

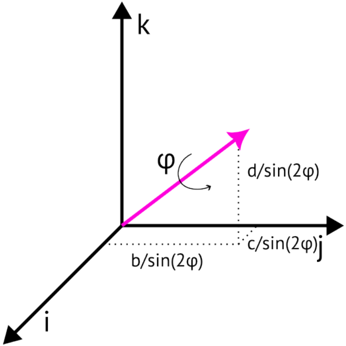


$$p = cos(2\phi) + sin(2\phi)(\frac{b}{sin(2\phi)}\mathbf{i} + \frac{c}{sin(2\phi)}\mathbf{j} + \frac{d}{sin(2\phi)}\mathbf{k})$$

Transformations

Quaternions

$$a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$



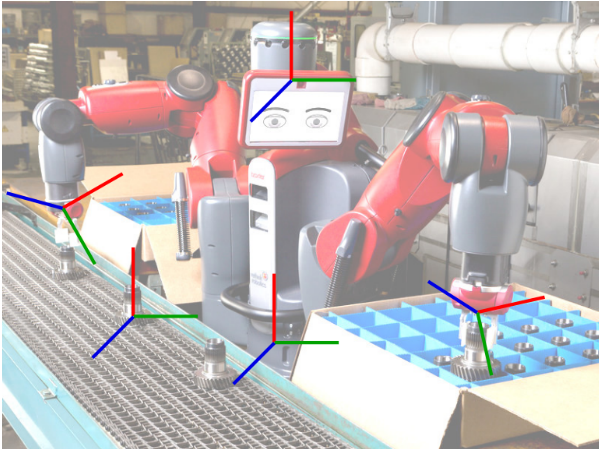
$$p = cos(2\phi) + sin(2\phi)(\frac{b}{sin(2\phi)}\mathbf{i} + \frac{c}{sin(2\phi)}\mathbf{j} + \frac{d}{sin(2\phi)}\mathbf{k})$$

$$p^{'} = qpq^-1$$

Switching frames

Because switching is useful

Sometimes, we know the coordinates of a point in one coordinate frame, but we need to describe it in a second frame. This is possible if we know the relative position of the two coordinate frames

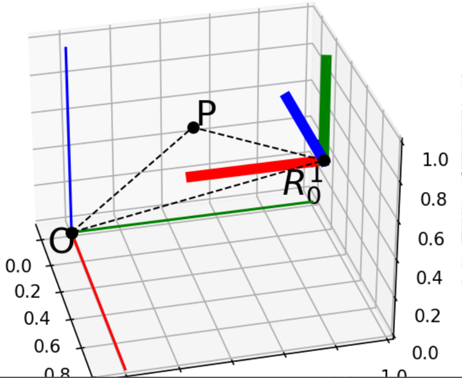


Switching frames

Because switching is useful

Sometimes, we know the coordinates of a point in one coordinate frame, but we need to describe it in a second frame. This is possible if we know the relative position of the two coordinate frames

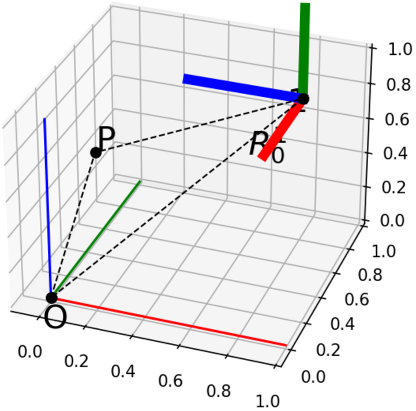
$$P_0 = R_0^1 P_1$$



Switching frames

Example

$$P_0 = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

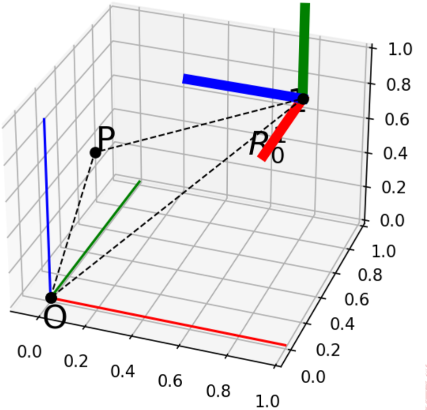


Switching frames

Example

$$P_0 = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

$$R_1^0 = \begin{bmatrix} 0 & -1 & 0 & 0.8 \\ 0 & 0 & 1 & -0.8 \\ -1 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



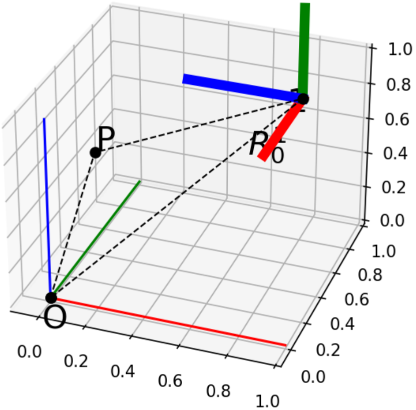
Switching frames

Example

$$P_0 = \begin{bmatrix} 0 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}$$

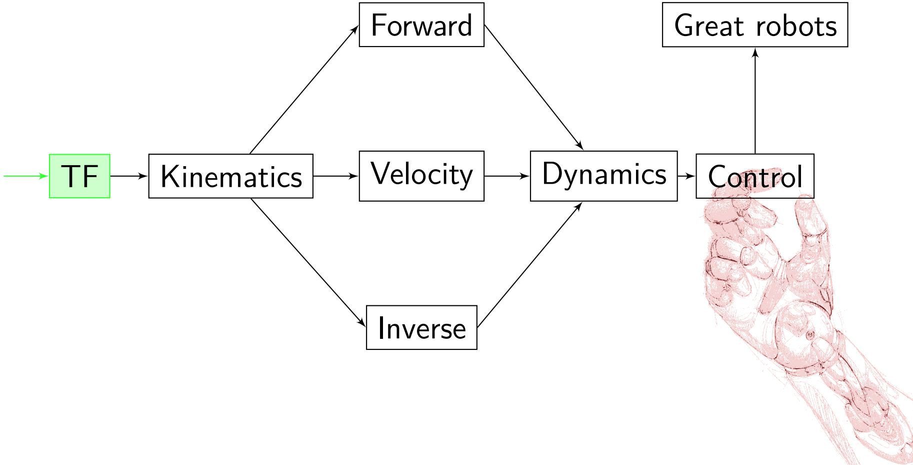
$$R_1^0 = \begin{bmatrix} 0 & -1 & 0 & 0.8 \\ 0 & 0 & 1 & -0.8 \\ -1 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1 = R_1^0 P_0 = \begin{bmatrix} 0.3 \\ -0.3 \\ 0.8 \\ 1 \end{bmatrix}$$



Grand scheme

The big picture



TF

Kinematics

Forward

Velocity

Inverse

Dynamics

Control

Great robots

Questions?