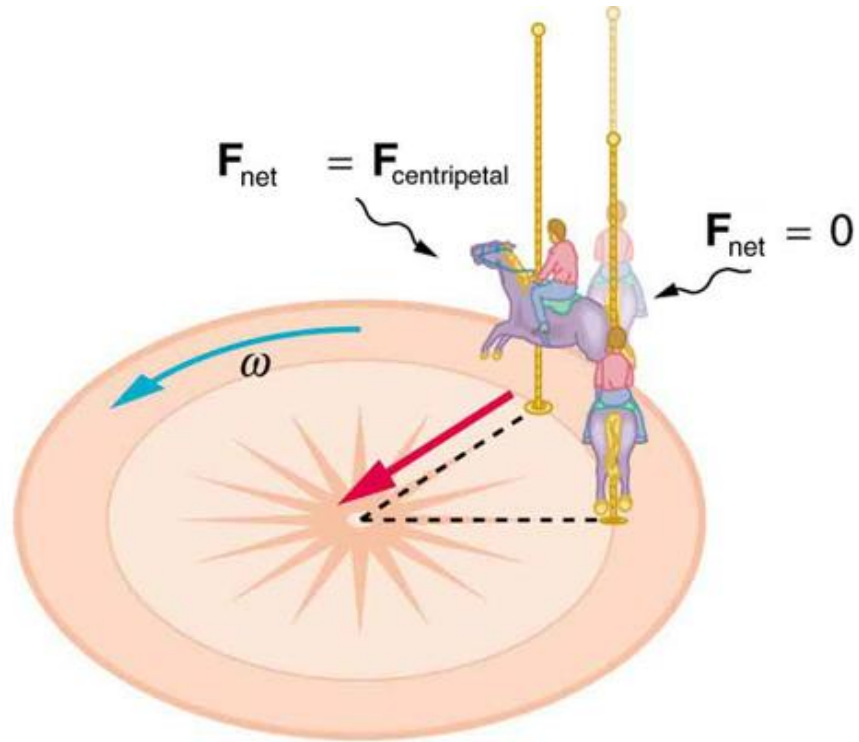

TRANSFORMATION FRAMES

ROBOTICS

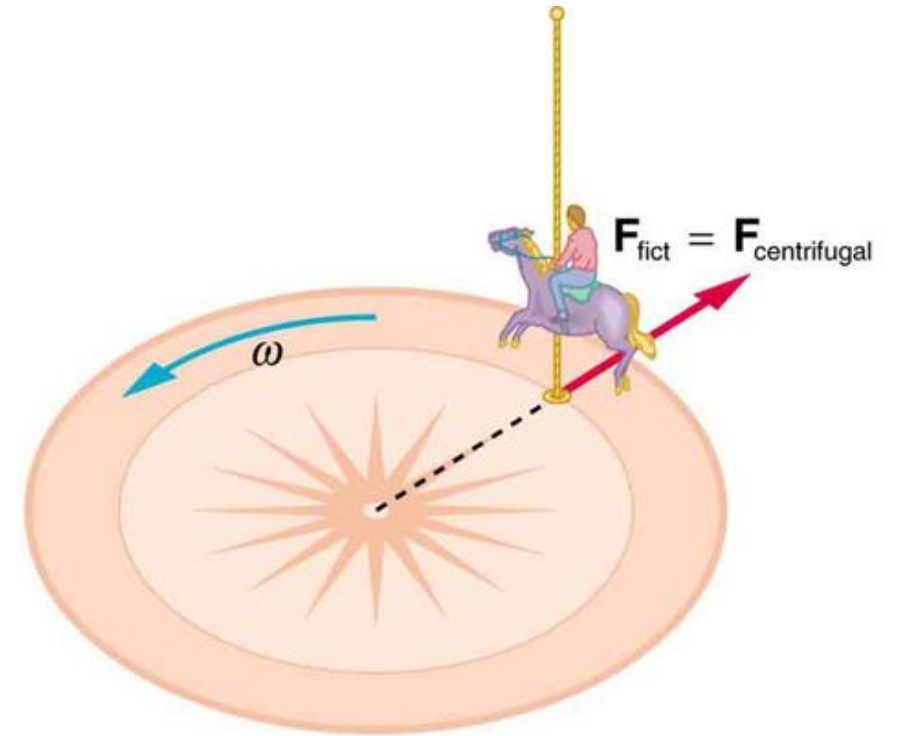


POLITECNICO
MILANO 1863

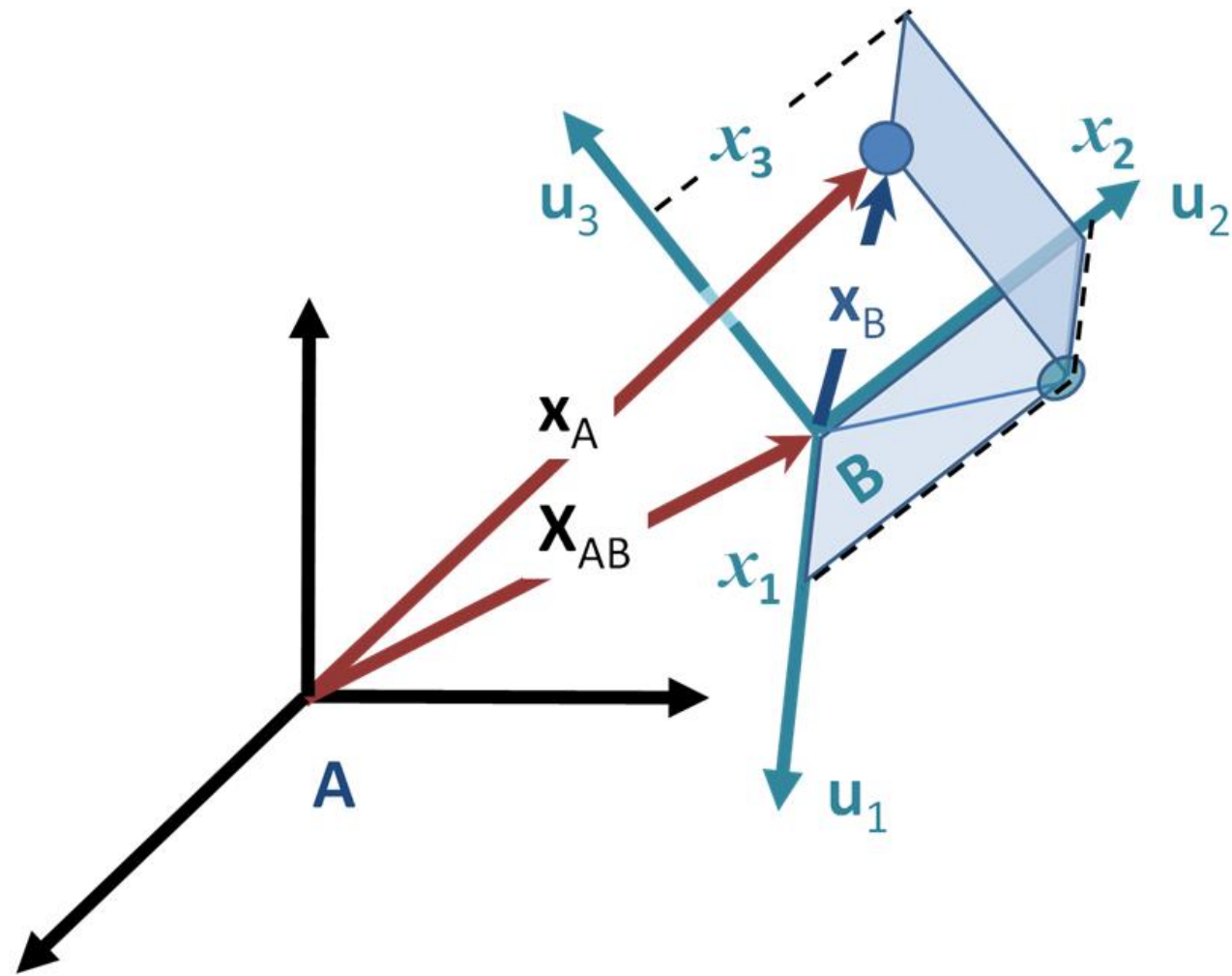
IN PHYSICS: AN EXAMPLE



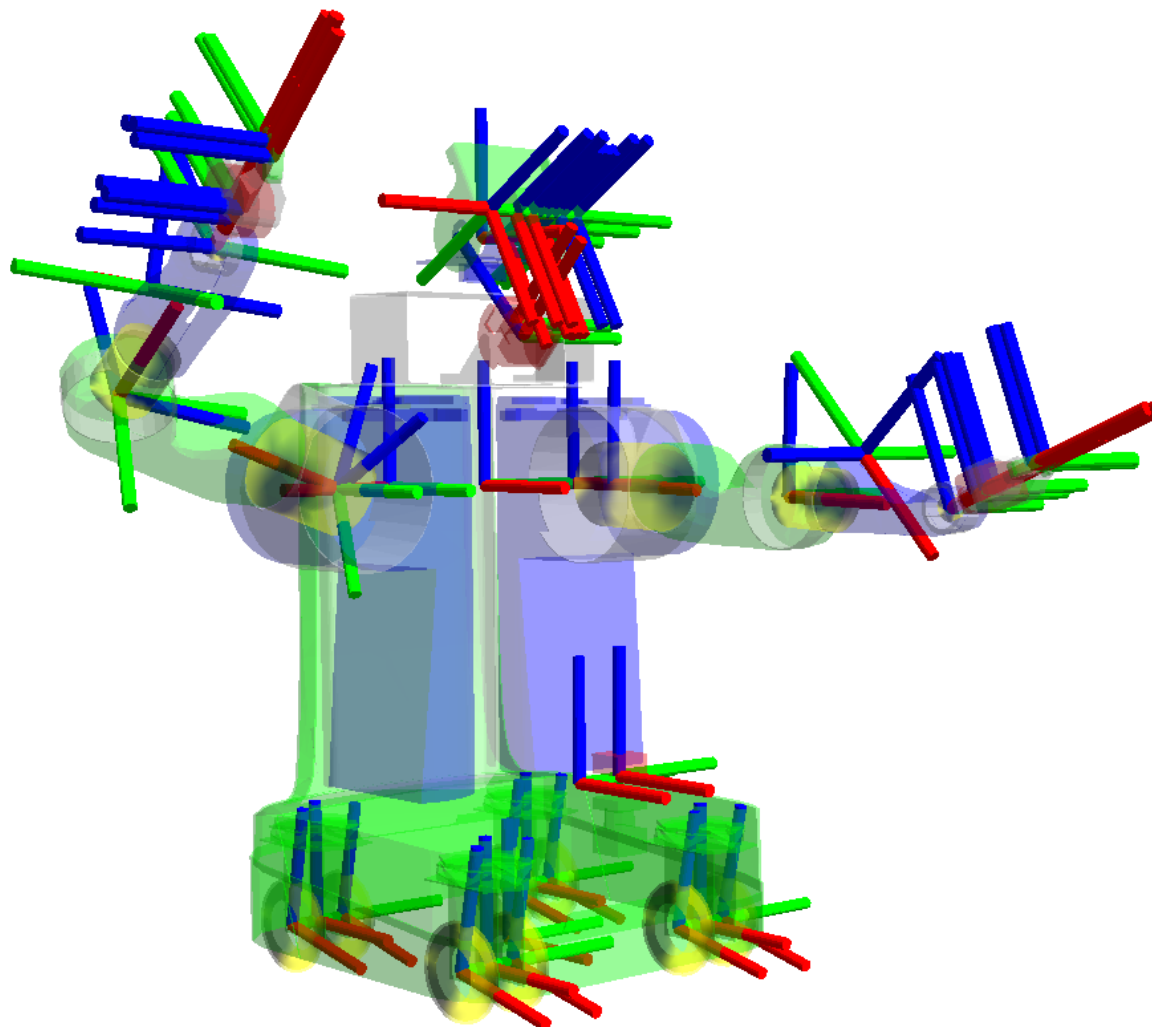
VS



TRANSFORMATION BETWEEN FRAMES



IN ROBOTICS





For manipulators:

A moving reference frame for each joint

A base reference frame

A world reference frame

For autonomous vehicles:

A fixed reference frame for each sensor

A base reference frame

A world reference frame

A map reference frame

The frames are described in a tree and each frame comes with a transformation between itself and the father/child

The world frame is the most important, but the others are used for

FROM ONE FRAME TO ANOTHER



How is it possible to convert from a frame to another? **Math**, lot of it.

In a tree of reference frames:

Define a roto-translation between parent and child

Combine multiple roto-translation to go from the root to the

ROTO-TRANSLATION



$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_y = \begin{bmatrix} \cos \beta & 0 & -\sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_z = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = R_x R_y R_z T$$



When the full transformation tree is available

Does all the hard work for us!

Interpolation, transformation, tracking

Keep track of all the dynamic transformation for a limited period of time

Decentralized

Provides position of a point in each possible reference frame

HOW TO ACHIEVE THIS MAGIC



Based on a broadcaster/listener paradigm

Create a broadcaster for each static transform (i.e. sensors)

Create a broadcaster for each dynamic transform (i.e. the robot)

Use a listener to collect interesting points (i.e. position of a mobile camera)

It is necessary to maintain the entire transformation tree

STATIC BROADCASTER



Used to broadcast a transformation that will never change in time

Command

```
static_transform_publisher x y z qx qy qz qw frame child_frame period_in_ms  
static_transform_publisher 0.7 -0.46 0.88 0 0 0 1 /base_link /gps 500
```

Embedded in the launch file

```
<node name="gps_transformation_publisher" pkg="tf"  
  type="static_transform_publisher"  
  args="0.7 -0.46 0.88 0 0 0 1 /base_link /gps 500" />
```

DYNAMIC BROADCASTER



```
#include <tf/transform_broadcaster.h>    //tf package in CMakeLists.txt

tf::TransformBroadcaster br;    //The broadcaster
tf::Transform transform;        //The transformation

transform.setOrigin(tf::Vector3(x, y, 0.0));
tf::Quaternion q(qx, qy, qz, qw);
transform.setRotation(q);
br.sendTransform(tf::StampedTransform(transform, ros::Time::now(), "world",
    "base_link"));
```

LISTENER



```
#include <tf/transform_listener.h>

tf::TransformListener listener;
tf::StampedTransform transform;

try {
    listener.waitForTransform("/world", "/base_link", ros::Time(0),
        ros::Duration(2.0));
    listener.lookupTransform("/world", "/base_link", ros::Time(0), transform);
} catch (tf::TransformException ex) { ROS_ERROR("%s", ex.what()); }
```

TFTOOLS



```
roslaunch tf tf_monitor
```

Gives a live overview of the current state of the tf tree

```
roslaunch tf view_frames
```

Creates a graphical representation of the current tf tree in the format of a PDF file

```
roslaunch tf tf_echo <source_frame> <target_frame>
```

Print information about a particular transformation between a `source_frame` and a `target_frame`