Representing Robot Pose: The good, the bad, and the ugly.

June 9, 2014

I was at ICRA last week and I had the pleasure to talk at the workshop entitled "What Sucks in Robotics and How to Fix It: Lessons Learned from Building Complex Systems" organized by Gian Diego Tipaldi and Cyrill Stachniss. This was a truly exceptional event where people showed up to talk honestly and challenge each other about robotics as a science and an engineering discipline.

My presentation was called "Representing Robot Pose: The good, the bad, and the ugly", where I tried to give some basic recommendations on how to talk about, write about, and code with representations of robot attitude and pose. I have seen many hours of productive time lost within my lab when students were trying to interface an open source package or process a dataset where the notion of robot pose was not clear enough to implement it correctly without trial and error (and pain and misery and suffering).

Here are the slides from the talk. To make everything easily accessible, I've written a brief overview below. Please let me know if anything is unclear of if you find typos!

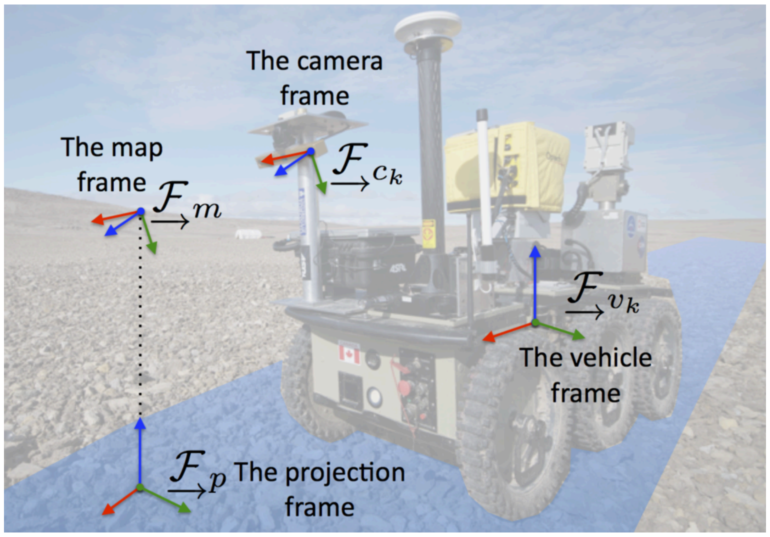
Overview

A large part of the practice of robotics is fundamentally about developing machines that can perceive and interact with the real physical world. And for that, we need to talk about, write about, write computer programs that represent **robot poses.** For the most part, we are extremely bad at this. There are hundreds of ways of turning a handful of scalars into a transformation matrix and, unless we are extremely clear about how to do this and how the result can be applied, we are condemning each other to wasted time and effort experimenting with the different possibilities until we find the one that fits. Rather than get in to religious discussions about which representation is correct, I propose a minimum amount of documentation to avoid ambiguity.

My goal in writing this is to convince you to go back to your documentation, papers, datasets, and open-source software to update the text so that the interpretation of the frame transformations is completely unambiguous to the user. Said another way, **I would like you to help me reduce the amount of suffering in the world.** There are many, many interesting problems that we still need to solve in robotics, and it pains me to see students and engineers losing days struggling with basics that can be avoided with some simple clear documentation.

I will use the notation that we have proposed for our software library "Kindr". We traced this notation back to the book "Elastic Multibody Dynamics -- A Direct Ritz Approach" by H. Bremer. I'll try to find time to make another post about notation that covers the major styles used in robotics.

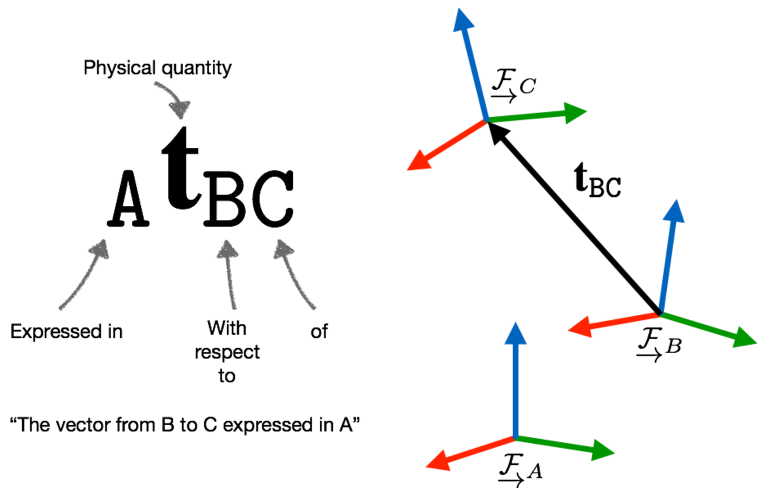
1. Always provide a frame diagram


The camera frame
$\underrightarrow{\mathcal{F}}_{c_k}$
The map frame
$\underrightarrow{\mathcal{F}}_{m}$
$\underrightarrow{\mathcal{F}}_{v_k}$
The vehicle frame
$\underrightarrow{\mathcal{F}}_{p}$
The projection frame

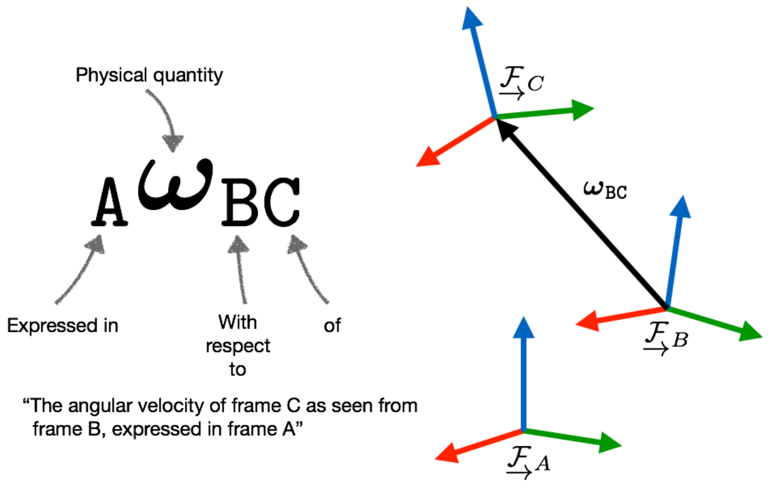
Any documentation about frame transformations is reliant on a good frame diagram that shows how the frames are placed on the robot. I follow the convention that the frames are colored with x=red, y=green, and z=blue. If you don't provide a frame diagram, it will be completely unclear to anyone how those frames are situated on the robot. Look at this wikipedia page for a selection of the widely varying conventions for terrestrial and aerospace vehicles.

2. When discussing vectors, be clear about what vectors you are providing

Al Kelly's tech report on wheeled kinematics is a good introduction to why you need three elements of decoration to clearly specify the coordinates of a vector quantity. Here's a pictorial cheat sheet:


Physical quantity
$_A\mathbf{t}_{BC}$
Expressed in
With respect to
of
"The vector from B to C expressed in A"
$\underset{\rightarrow}{\mathcal{F}}_C$
$\mathbf{t}_{BC}$
$\underset{\rightarrow}{\mathcal{F}}_B$
$\underset{\rightarrow}{\mathcal{F}}_A$

I find that many people are not convinced that we always need three pieces of decoration. I think angular velocity is a good motivator:


Physical quantity
$_A\boldsymbol{\omega}_{BC}$
Expressed in
With respect to
of
"The angular velocity of frame C as seen from frame B, expressed in frame A"
$\underset{\rightarrow}{\mathcal{F}}_C$
$\boldsymbol{\omega}_{BC}$
$\underset{\rightarrow}{\mathcal{F}}_B$
$\underset{\rightarrow}{\mathcal{F}}_A$

These kinds of terms actually come up when computing the equations for wheeled or manipulator kinematics and it ends up extremely useful to have notation expressive enough to be able to write them down succinctly.

3. When discussing orientation, be very clear about what orientation you are

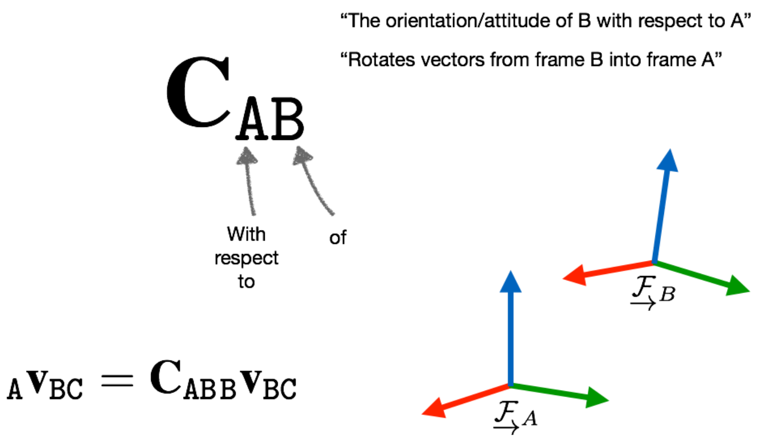
providing

There are two big points to consider here. First, make sure you specify how to go from whatever scalars you provide to a full rotation matrix. Second, clearly specify which rotation matrix this is (world-to-body, or body-to-world, for example). This will be most clear if you provide some text like this:

The resulting rotation matrix, $\mathbf{C}_{\mathtt{WB}}$, represents the orientation of the robot body frame, $\underrightarrow{\mathcal{F}}_{\mathtt{B}}$, with respect to the world frame, $\underrightarrow{\mathcal{F}}_{\mathtt{W}}$, such that a vector expressed in the body frame, ${}_{\mathtt{B}}\mathbf{v}$, can be rotated into the world frame by

$$ {}_{\mathtt{W}}\mathbf{v} = \mathbf{C}_{\mathtt{WB}\,\mathtt{B}}\mathbf{v}. \qquad (1) $$

It is also good to include some suggested phrases in your documentation and use these phrases throughout your work. Here is a pictorial cheat sheet with some suggested phrases. *Note that there is no agreement on these phrases across robotics so please don't expect that other people use them the way that I do!*


"The orientation/attitude of B with respect to A"
"Rotates vectors from frame B into frame A"
$\mathbf{C}_{\mathtt{AB}}$
With respect to
of
$\underrightarrow{\mathcal{F}}_{B}$
$\underrightarrow{\mathcal{F}}_{A}$
${}_{\mathtt{A}}\mathbf{v}_{\mathtt{BC}} = \mathbf{C}_{\mathtt{AB}\,\mathtt{B}}\mathbf{v}_{\mathtt{BC}}$

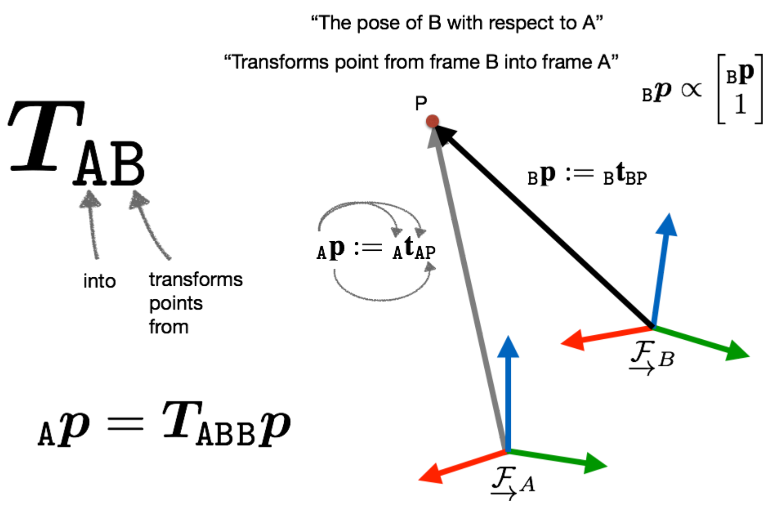
4. When discussing pose, be very clear about what pose you are providing

This advice will be almost exactly the same as what I gave for rotation matrices above. Again, there are two big points to consider. First, make sure you specify how to go from whatever scalars you provide to a full transformation matrix. Second, clearly specify which transformation matrix this is. Here is my suggested text:

The resulting transformation matrix, $\boldsymbol{T}_{\mathtt{WB}}$, represents the pose of the robot body frame, $\underrightarrow{\mathcal{F}}_{\mathtt{B}}$, with respect to the world frame, $\underrightarrow{\mathcal{F}}_{\mathtt{W}}$, such that a point expressed in the body frame, ${}_{\mathtt{B}}\mathbf{p}$, can be transformed into the world frame by

$$ {}_{\mathtt{W}}\boldsymbol{p} = \boldsymbol{T}_{\mathtt{WB}\,\mathtt{B}}\boldsymbol{p}. \qquad (1) $$

I've again included some suggested phrases in the pictorial example.



"The pose of B with respect to A"
"Transforms point from frame B into frame A"
T_{AB}
into
transforms points from
$_{A}\boldsymbol{p} = T_{AB\,B}\boldsymbol{p}$
P
$_{B}\boldsymbol{p} \propto \begin{bmatrix} _{B}\mathbf{p} \\ 1 \end{bmatrix}$
$_{B}\mathbf{p} := {}_{B}\mathbf{t}_{BP}$
$_{A}\mathbf{p} := {}_{A}\mathbf{t}_{AP}$
$\underrightarrow{\mathcal{F}}_{B}$
$\underrightarrow{\mathcal{F}}_{A}$

I have a few more notes for this example.

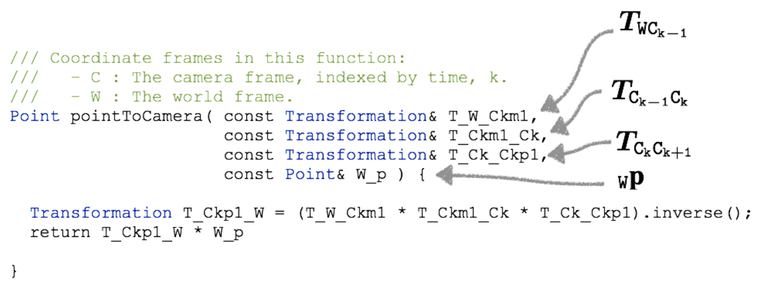
1. Once again, there is no agreement on these phrases so please don't assume that people use them the same way.
2. I've introduced simplified notation for points with only one left subscript. If your paper or dataset is just dealing with points and poses, this can simplify the notation greatly.
3. Since my PhD thesis, I have used homogeneous coordinates to represent points. To disambiguate between homogeneous coordinates (4x1) and Euclidean coordinates (3x1), I use a different font; homogeneous are bold italics and Euclidean are bold upright.

5. If it is not 100% clear, have fear. Use a 3D plotting tool to check your guess

Most times it will not be totally clear how to go from the scalars provided to a transformation matrix. If you are not 100% sure of the answer, the least painful thing to do is to fire up your favorite 3D plotting tool and plot the answers until things look right. Then write the author and explain to them what documentation you would have needed to get it right without guesswork.

Suggestions for coding style

At the autonomous systems lab, we try to transfer the notation you see above directly into code so that it is clear when multiple people work on the same function.



```
/// Coordinate frames in this function:
///   - C : The camera frame, indexed by time, k.
///   - W : The world frame.
Point pointToCamera( const Transformation& T_W_Ckm1,
                     const Transformation& T_Ckm1_Ck,
                     const Transformation& T_Ck_Ckp1,
                     const Point& W_p ) {

  Transformation T_Ckp1_W = (T_W_Ckm1 * T_Ckm1_Ck * T_Ck_Ckp1).inverse();
  return T_Ckp1_W * W_p

}
```

The important points are:

1. Defining a standard notation in code for transformation matrices, points, and vectors.
2. Commenting each function or class with the list of coordinate frames used.
3. adopting some notation for time indices. Here I have used "kp1" for k+1 and "km1" for k-1.

Great! Please leave me any comments or questions below.

Share   26 Likes

NEWS   NEWER   OLDER

Comments (4)

Newest First

Preview Post Comment…

Martin Weisenhorn

2 years ago · 0 Likes

It is a while ago since you wrote this now. But the situation in many software packages e.g. OpenCV is still the same, unfortunately. Your solution is exactly what it takes in my opinion. I wonder how engineers seem to master rotations and translations without a clear notation.

Aditya 2 years ago · 0 Likes

I completely agree. I don't think anyone tries to understand the details of transformations which ultimately results in "buggy" code. And then life eventually becomes miserable.

quaternion 4 years ago · 0 Likes

Thank you for the wonderful article. What is your favorite 3D plotting tool to use for visualization when in doubt?

Lin 7 years ago · 0 Likes

Good post, but I think notation defined here is different from some other papers/books. I wonder if you have tried to write a post about notation that covers the major styles used in robotics? Thanks a lot.