CS 3630 Lecture Notes on Pose and Orientation

Frank Dellaert

January 17, 2014

1 Geometry

These notes I created last year might provide some additional perspective. Please note I use a slightly different notation from Corke: below I write R_A^B to denote (for example) a rotation matrix that transforms points in frame A to points in frame B, whereas Corke would write ${}^B R_A$. Also, at the end I show that it is possible to define composition without 3×3 matrix multiplication: there are many ways to represent the same transformations, and it simply requires one to define composition accordingly.

1.1 Points and Translations in 2D

We represent points in 2D as $p^A \in \mathbb{R}^2$, where the superscript index denotes a coordinate frame, in this case the frame A.

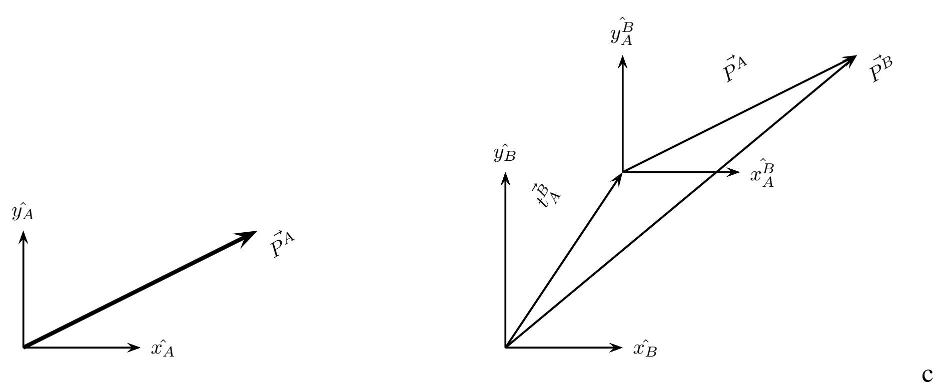

Figure 1: Translation of a point p^A to a point p^B.

A **translation** $t_A^B \in \mathbb{R}^2$ transforms points in frame A to a translated frame B, which is done simply by vector addition:

$$p^B = t_A^B + p^A$$

where the indices on t_A^B indicate the source and destination frames. This is illustrated in Figure 1.

1.2 Rotations in 2D

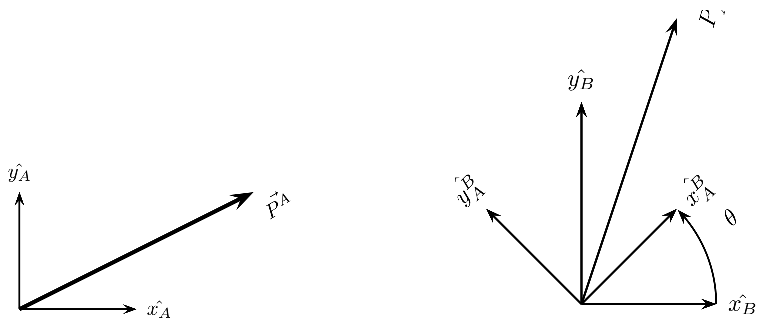

Figure 2: Rotation of a point p^A to a point p^B.

Rotating a point in 2D around the origin from a frame A to a rotated frame B can be done by multiplying with a 2×2 orthonormal rotation matrix

$$p^B = R_A^B p^A$$

where the indices on R_A^B indicate the source and destination frames, and

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

for a **rotation** by an angle θ. This is illustrated in Figure 2.

1.3 Robot Pose: Rigid 2D Transformations

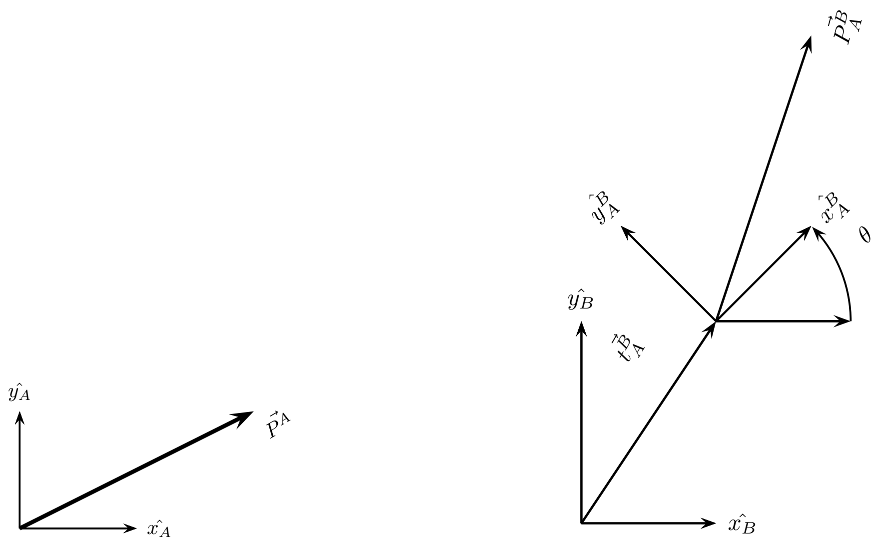

Figure 3: A rigid 2D transformation applied to the point p^A.

The pose ξ_R^W of a robot in the world frame consists of *three* numbers:

$$\xi_R^W = \begin{bmatrix} t_x \\ t_y \\ \theta \end{bmatrix} = (t, R(\theta))$$

where we can use either a three-vector notation or the translation-rotation tuple notation as above.

We can transform a 2D point p^R in the robot frame to the world frame by

$$p^W = R_R^W p^R + t_R^W$$

as illustrated in Figure 3. For example, you can use this to calculate the world coordinates of a sensor on the robot body, for which only have the *relative* (robot-frame) coordinates.

Composition can be understood by examining the result of transforming a point twice:

$$p^C = R_B^C \left(R_A^B p^A + t_A^B\right) + t_B^C = \left(R_B^C R_A^B\right) p^A + \left(R_B^C t_A^B + t_B^C\right)$$

In short, the operation for **composi**ng two rigid 2D transformations can be defined as:

$$\xi_A^C = \xi_B^C \oplus \xi_A^B = \left(R_B^C,\, t_B^C\right) \oplus \left(R_A^B,\, t_A^B\right) \triangleq \left(R_B^C R_A^B,\, R_B^C t_A^B + t_B^C\right) \tag{1}$$