Laser beacons and dead reckoning

Florian Reinhard *Pius von Däniken*
`florian.reinhard@epfl.ch` `pius.vondaeniken@epfl.ch`

Contents

1 Introduction

1.1 The Problem

We want to calculate the position of a moving robot in a 2D plane by measuring the relative angles between three beacons with known coordinates. As we don't measure the absolute angles to some fixed direction (like north in nautical bearing [2]), we absolutely need all three angles to calculate the Cartesian coordinates of our robot.

If the position of the three beacons and the robot all lie on a circle, the fact that the transformation from the measured angles to the coordinates is not *injective*, starts to present a problem as described in Section 1.2.2 and shown in Figure 1.

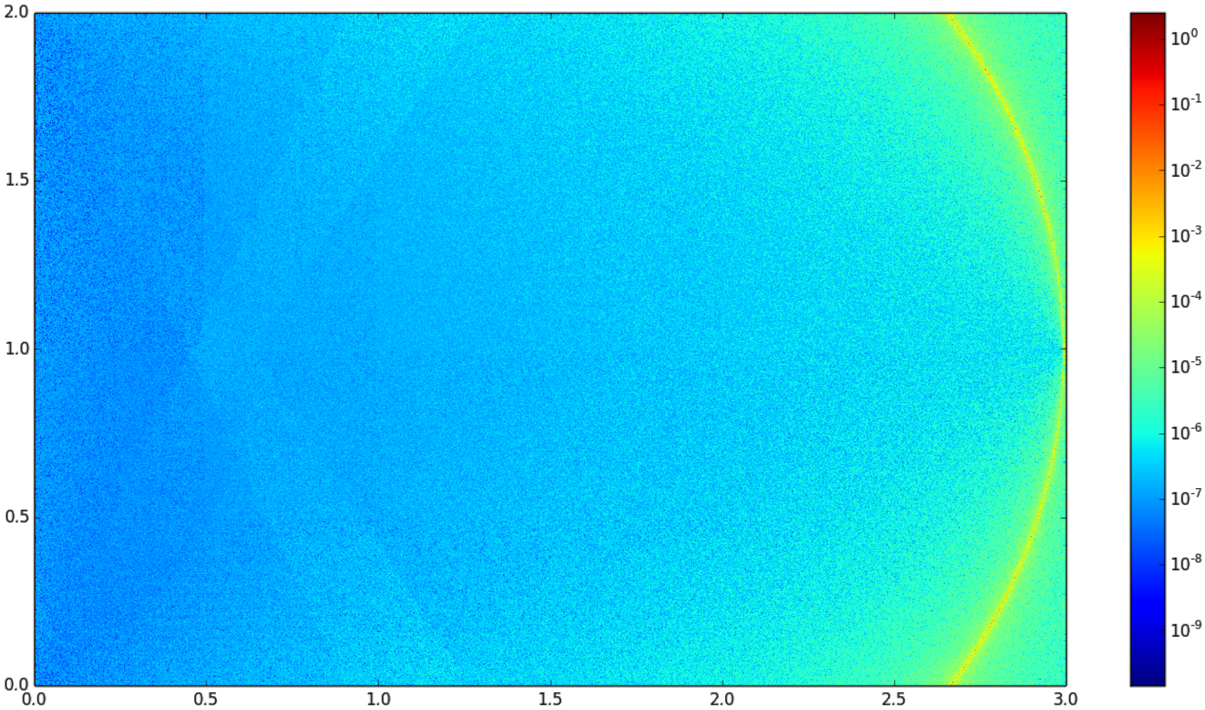

Fig. 1: Representation of the magnitude of the error due to *floating point errors* errors when calculating the position from the measured angles. Every pixel represents a position that has been transformed into the angles that should have been measured and then back into cartesian coordinates. The *beacons* are positioned at $(0, 0)$, $(0, 2)$, and $(3, 1)$

1.2 Positioning with beacons

We measure the angles x, y, and z and we want to calculate the vector P. It seems like a logical conclusion to use a *barycentric coordinate system* (1.2.1) to solve this problem.

1.2.1 Barycentric coordinates

In a two-dimensional barycentric coordinate system a position is specified as the center of mass of masses placed at the vertices of a triangle. In our case the vertices are at the beacons' positions.

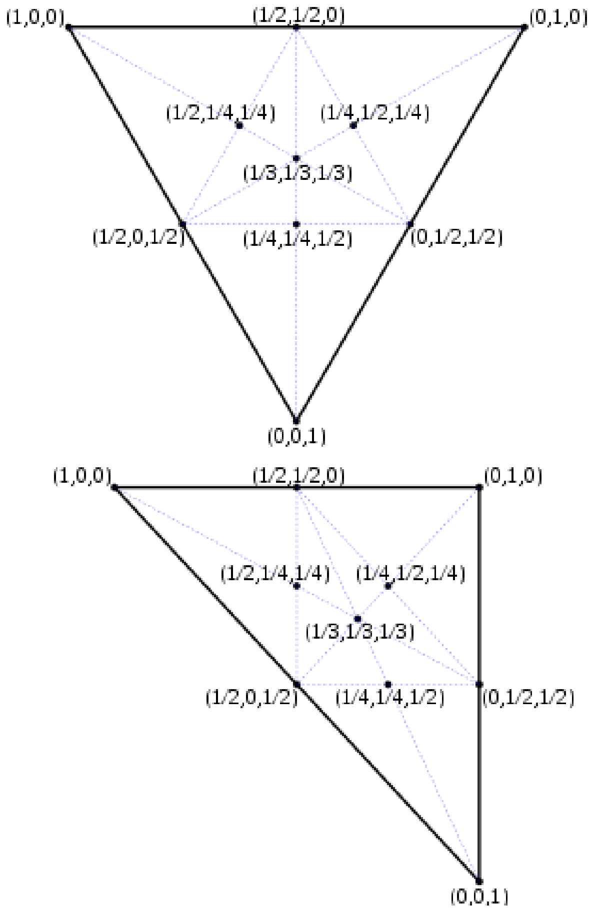

Fig. 2: Several points in different barycentric coordinate systems.

1.2.2 The algorithm

The *barycentric coordinates* of P in figure 3 are

$$\left(\frac{1}{\cot A - \cot x} : \frac{1}{\cot B - \cot y} : \frac{1}{\cot C - \cot z}\right) \tag{1}$$

where A, B, and C are the triangle's angles at the corresponding vertices [1].

If now P lies on the circle going through A, B, and C (figure 4), we have the problem that two of the three coordinates in equation 1 are equal to $\frac{1}{0}$ and thus tend to infinity. On top of that, the coordinates are constant per segment between vertices (this can be explained by the *inscribed angle theorem* [3]).

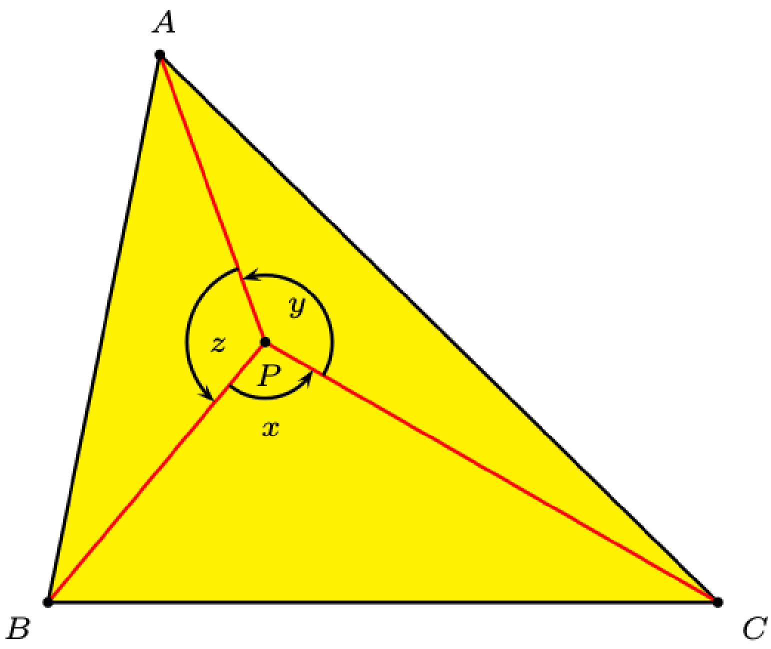

Fig. 3: The three angles that are measured when positioning with beacons.

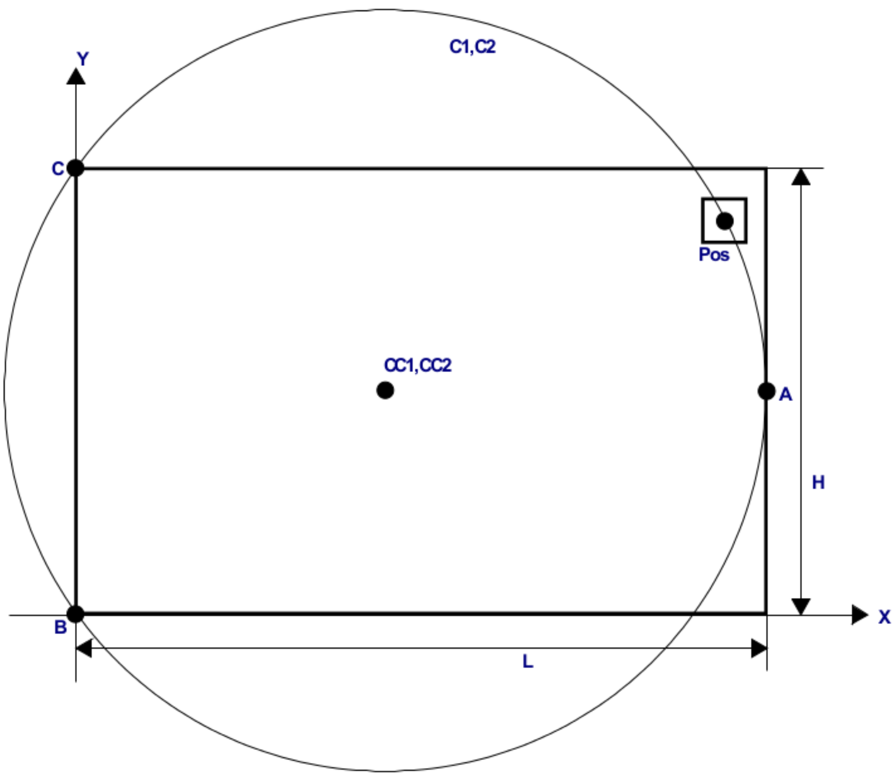

Fig. 4: P lies on the circle going through A, B, and C.

1.3 Dead reckoning

Dead reckoning is the process of deducing the current position off of a previously determined position and the advanced distance based upon known or estimated speeds over elapsed time and course. [4]

1.4 Kalman Filter

Here is just the algorithm adapted to our needs. Maybe start with *wikipedia* [5] for actual explanations.

Predict

$$\hat{\mathbf{x}}_k = \mathbf{F}_k \mathbf{x}_{k-1} \tag{2}$$

$\hat{\mathbf{x}}_k$ is the new *prediction* of the state, $\mathbf{F}_k$ and $\mathbf{x}_{k-1}$ are the *state transition matrix* and the old state estimation.

$$\hat{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F^T}_k + \mathbf{Q}_k \tag{3}$$

The covariance $\hat{\mathbf{P}}_k$ of the predicted state depends on the previous covaraiance, the state transition function and the *state transition error* $\mathbf{Q}_k$ (see 2.2.1).

Update

$$\mathbf{y}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k \tag{4}$$

$\mathbf{y}_k$ is the *measurement residual*, $\mathbf{z}_k$ is the measurement, and $\mathbf{H}_k$ transforms the state in to *measurement space*. (Note that we don't have a control.)

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k \mathbf{H^T}_k + \mathbf{R}_k \tag{5}$$

$\mathbf{S}_k$ is the *residual covariance* and $\mathbf{R}_k$ is the covariane of the measurement's *noise*.

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H^T}_k \mathbf{S^{-1}}_k \tag{6}$$

$\mathbf{K}_k$ is the *Kalman gain*.

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}_k \mathbf{y}_k \tag{7}$$

$\mathbf{x}_k$ is the *updated state estimation*.

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\, \mathbf{P}_k \tag{8}$$

$\mathbf{P}_k$ is the *updated estimate covariance*.

2 Adding dead reckoning to the game

2.1 Why?

Because of the enormous error of the beacon system at certain positions and the high probability of outages (another robot obstructing the line of sight), a *Kalman filter* integrating dead reckoning and the beacons should increase the precision and usability of the system.

2.2 Kalman Filter

State variables A possible state for the Kalman filter is

$$\mathbf{x_k} = \begin{pmatrix} x_k \\ y_k \\ x_{k-1} \\ y_{k-1} \end{pmatrix} \tag{9}$$

State transition

$$\mathbf{F_k} = \begin{pmatrix} 1+\tau & 0 & -\tau & 0 \\ 0 & 1+\tau & 0 & -\tau \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{10}$$

Where τ is the factor $\frac{t_k - t_{k-1}}{t_{k-1} - t_{k-2}}$. This makes for the following computation:

$$\mathbf{x_k} = \mathbf{x_{k-1}} + \mathbf{\Delta x_k} = \mathbf{x_{k-1}} + \mathbf{v_{k-1}}\Delta t_k = \mathbf{x_{k-1}} + \frac{\mathbf{x_{k-1}} - \mathbf{x_{k-2}}}{\Delta t_{k-1}}\Delta t_k \tag{11}$$

Measurement

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{12}$$

Not much to say about that.

2.2.1 Variances

For the state transition we can safely assume that almost 100% of the time, the robot won't accelerate more than the maximal acceleration which our regulation will use (only collisions and such can make the robot to accelerate more). For a *gaussian distribution*, 99.8% are inside 4σ, so we say that $4\sigma = \frac{1}{2}a_{\max}(t_k - t_{k-1})^2$ and thus $\sigma = \frac{1}{8}a_{\max}(t_k - t_{k-1})^2$ is the standard deviation of our control update i.e. **Q**.

$$\mathbf{Q_k} = \begin{pmatrix} \frac{1}{64}a_{\max}^2\Delta t^4 & \frac{1}{64}a_{\max}^2\Delta t^4 & 0 & 0 \\ \frac{1}{64}a_{\max}^2\Delta t^4 & \frac{1}{64}a_{\max}^2\Delta t^4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{13}$$

The measurement 's variance will have to determined experimentally and should be adapted when the robot approaches the *circle of death*.

2.3 Next step: EKF

Because the transform described in equation 1 isn't linear, it's actually quite wrong to use a normal *Kalman filter* for the measurement.

Measurement update with *Extended Kalman Filter* What we measure with our beacons isn't actually a position, but relative angles between beacons. So, our $\mathbf{H_k}$ should actually be $h_k(x_k)$, a function that transforms the current state to the measurement space, i. e. a position to angles seen between beacons from this position. For two beacons A and B at positions $\begin{pmatrix} A_x \\ A_y \end{pmatrix}$ and $\begin{pmatrix} B_x \\ B_y \end{pmatrix}$ you see the angle

$$\alpha = \tan^{-1}\left(\frac{B_y - y}{B_x - x}\right) - \tan^{-1}\left(\frac{A_y - y}{A_x - x}\right) \tag{14}$$

at the position $\begin{pmatrix} x \\ y \end{pmatrix}$. This is just the difference of the absolute angles of $\overline{PA}$ and $\overline{PB}$.

To be even more exact, we don't measure angles directly, but the time between between the passage of the laser beam at two different beacons. Given a known rotational speed ω of the beam, we write

$$\Delta t_{AB} = \frac{\tan^{-1}\left(\frac{B_y - y}{B_x - x}\right) - \tan^{-1}\left(\frac{A_y - y}{A_x - x}\right)}{\omega} \tag{15}$$

.

Like that we can do a measurement update for each of the three angles:

$$h_{AB_k}\left(x_k, y_k, \omega_k\right) = \frac{1}{\omega_k}\left(\tan^{-1}\left(\frac{B_y - y_k}{B_x - x_k}\right) - \tan^{-1}\left(\frac{A_y - y_k}{A_x - x_k}\right)\right) \tag{16}$$

$$h_{BC_k}\left(x_k, y_k, \omega_k\right) = \frac{1}{\omega_k}\left(\tan^{-1}\left(\frac{C_y - y_k}{C_x - x_k}\right) - \tan^{-1}\left(\frac{B_y - y_k}{B_x - x_k}\right)\right) \tag{17}$$

$$h_{CA_k}\left(x_k, y_k, \omega_k\right) = \frac{1}{\omega_k}\left(\tan^{-1}\left(\frac{A_y - y_k}{A_x - x_k}\right) - \tan^{-1}\left(\frac{C_y - y_k}{C_x - x_k}\right)\right) \tag{18}$$

The *EKF* uses the *Jacobian* of $h_k\left(\mathbf{x_k}\right)$ and thus we need it's *partial derivatives*:

$$\frac{\partial}{\partial x_k} h_{AB_k}\left(x_k, y_k, \omega_k\right) = \frac{1}{\omega_k}\left(\frac{B_y - y}{\left(B_x - x\right)^2 + \left(B_y - y\right)^2} - \frac{A_y - y}{\left(A_x - x\right)^2 + \left(A_y - y\right)^2}\right) \tag{19}$$

$$\frac{\partial}{\partial y_k} h_{AB_k}\left(x_k, y_k, \omega_k\right) = \frac{1}{\omega_k}\left(\frac{A_y - y}{\left(A_x - x\right)^2 + \left(A_y - y\right)^2}\right) - \frac{B_y - y}{\left(B_x - x\right)^2 + \left(B_y - y\right)^2} \tag{20}$$

$$\frac{\partial}{\partial \omega_k} h_{AB_k}\left(x_k, y_k, \omega_k\right) = -\frac{1}{\omega_k^2}\left(\tan^{-1}\left(\frac{B_y - y_k}{B_x - x_k}\right) - \tan^{-1}\left(\frac{A_y - y_k}{A_x - x_k}\right)\right) \tag{21}$$

Extended state You may have noticed that ω_k has been introduced as a new variable in the preceding paragraph. With the *EKF* we can keep track of the laser beam's rotational speed directly by adding it to the state:

$$\mathbf{x_k} = \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \\ \omega_k \end{pmatrix} \tag{22}$$

This would give us the following *state transition matrix*:

$$\mathbf{F_k} = \begin{pmatrix} 1 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{23}$$

Measuring $\omega_\mathbf{k}$ After every passage of a beacon, a measurement update on the rotational speed of the laser could be done by measuring the time between two passages of the same beacon.

Computation time *To be done.*

3 Results and expectations

To be done.

References

[1] Nikolaos Dergiades. `http://forumgeom.fau.edu/FG2009volume9/FG200921.pdf`.

[2] wikipedia.org. `http://en.wikipedia.org/wiki/Bearing_(navigation)`.

[3] wikipedia.org. `http://en.wikipedia.org/wiki/Inscribed_angle#Theorem`.

[4] wikipedia.org. `http://en.wikipedia.org/wiki/Dead_reckoning`.

[5] wikipedia.org. `http://en.wikipedia.org/wiki/Kalman_filter`.