Robot Calibration

Probabilistic Robotics Project

Alessandro Paglialunga

December 2023

1 Introduction

In this work, a traction drive wheel (aka front-tractor tricycle) was calibrated using an odometric trajectory based on sensor readings, representing the laser pose relative to its reference frame and initial pose.
In addition, the wheeled mobile robot was equipped with absolute and incremental encoders to estimate steering and drive (traction) velocities respectively.

The rest of the document is organized as follows: Sect. 2 reviews the robot kinematic model; Sect. 3 illustrates the least squares method for odometry calibration; Sect. 4 explains the code architecture and main procedures; Sect. 5 reports the results obtained and concludes the document.

2 Kinematic Model

The generalized coordinates used to represent the configuration of the robot are as follows $\mathbf{q} = (x, y, \theta, \psi)^T$ (figure 1), i.e., a set of generalized coordinates for the tricycle. In the following, the two-wheel rear axle is assimilated to a single wheel located at the axle midpoint, therefore robot has two wheels: front and rear wheels. The kinematic constraints acting on the robot are two (one "pure rolling" condition for each wheel):

$$\begin{aligned} \dot{x_f}\sin(\theta+\psi) - \dot{y_f}\cos(\theta+\psi) &= 0 \\ \dot{x}\sin(\theta) - \dot{y}\cos(\theta+\psi) &= 0 \end{aligned} \tag{1}$$

where (x_f, y_f) and (x, y) are the Cartesian coordinates of P_f (tricycle front wheel center) and P (tricycle rear wheel center), respectively.
Being

$$\begin{aligned} x &= x_f - l\cos(\theta) \\ y &= y_f - l\sin(\theta) \end{aligned} \tag{2}$$

where l is the length of the baseline or distance between the front wheel and the kinematic center, which is in the middle rear wheels axis. Therefore, kinematic constraints can be expressed in terms of x_f, y_f, where the encoders are located, as follows:

$$\begin{aligned} \dot{x_f}\sin(\theta+\psi) - \dot{y_f}\cos(\theta+\psi) &= 0 \\ \dot{x_f}\sin(\theta) - \dot{y_f}\cos(\theta) + \dot{\theta}l &= 0 \end{aligned} \tag{3}$$

or, in Pfaffian form

$$\begin{bmatrix} \sin(\theta+\psi) & \cos(\theta+\psi) & 0 & 0 \\ \sin(\theta) & -\cos(\theta) & l & 0 \end{bmatrix} \times \begin{bmatrix} \dot{x_f} \\ \dot{y_f} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{A}^T(\boldsymbol{q})\,\dot{\boldsymbol{q}} = \boldsymbol{0} \tag{4}$$

A basis for the null space of $\boldsymbol{A}^T$, or vectors field, can be easily obtained

$$\boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} \cos(\theta+\psi) & 0 \\ \sin(\theta+\psi) & 0 \\ \sin(\psi)/l & 0 \\ 0 & 1 \end{bmatrix} = (\boldsymbol{g}_1(\boldsymbol{q}),\, \boldsymbol{g}_2(\boldsymbol{q})) \tag{5}$$

where $\boldsymbol{g}_1(\boldsymbol{q})$ is driving vector field and $\boldsymbol{g}_2(\boldsymbol{q})$ is steering vector field. The kinematic control system is then

$$\dot{q} = \boldsymbol{g}_1(\boldsymbol{q})\, v + \boldsymbol{g}_2(\boldsymbol{q})\, \omega \tag{6}$$

where v and ω are respectively the driving and the steering velocity of the front-tractor tricycle, since an incremental encoder and an absolute encoder are both mounted on the front wheel for traction and steering, respectively. Accordingly, driving and steering velocities are measured as follows:

$$v = k_t \frac{\Delta I_t}{\Delta T\,(maxI - 1)} \tag{7}$$

$$\omega = -k_s 2\pi \frac{maxA - A_t}{\Delta T\, maxA} + steer_{off} \tag{8}$$

$$\omega = k_s 2\pi \frac{A_t}{\Delta T\, maxA} + steer_{off} \tag{9}$$

where ΔI_t is the difference between incremental encoder measure at time t and incremental encoder measure at time $t-1$, ΔT is the difference between

consecutive timestamps, k_s indicates how many radians correspond to one tick of absolute encoder, k_t represents how many meters correspond to one tick of incremental encoder and $steer_{off}$ is the angle to which the zero of the wheel corresponds.
Two different equations for ω are used to obtain an angle ψ. Equation (8) is used when the absolute encoder measurement is between $[0, \frac{maxA}{2})$ (where $maxA$ is the maximum value of the absolute encoder) and equation (9) is used when the encoder measurement is between $[\frac{maxA}{2}, maxA)$ this technique allows to obtain an angle ψ between $[-\pi, +\pi]$

Since the measurements, which are used in the least squares method to calibrate the robot, are laser poses, it is useful to analyze how the prediction of laser pose can be obtained from the front wheel configuration. Starting from the driving wheel the rear wheel position, where the kinematic center is located, can be easily obtained with equation (2). From here it will be necessary to consider the position of the laser and its orientation relative to the robot, such information can be contained within such a transformation matrix

$$\begin{bmatrix} \cos(laser_\theta) & -\sin(laser_\theta) & laser_x \\ \sin(laser_\theta) & \cos(laser_\theta) & laser_y \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The following matrix transformation represents the rigid offset between the robot and laser reference frames

$$\boldsymbol{T_{off}} = \begin{bmatrix} \cos(\theta_l^r) & -\sin(\theta_l^r) & pos_{l,x}^r \\ \sin(\theta_l^r) & \cos(\theta_l^r) & pos_{l,y}^r \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where $(pos_{l,x}^r, pos_{l,y}^r, \theta_l^r)$ is 2D laser position and orientation w.r.t robot reference frame. Whereas the robot pose with respect to its initial position $(0, 0, 0)$ and its reference frame can be represented by the following matrix transformation

$$\boldsymbol{T_R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & x \\ \sin(\theta) & \cos(\theta) & y \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

so the laser pose with respect to its initial position and its reference frame can be obtained starting from robot pose by the following matrix transformation

$$\boldsymbol{T_L} = \boldsymbol{T_{off}}\, \boldsymbol{T_L}\, \boldsymbol{T_{off}^{-1}} \tag{13}$$

from which it is possible to estimate (x_L, y_L, θ_L) that is the 2D laser position and its orientation w.r.t. its initial pose and its reference system[1].

This estimate is used as a prediction for the least square method and compared with the measurements in the dataset. The parameters to be calibrated are been introduced in the previous formulas:

- k_s: how many radians correspond to one tick of absolute encoder
- k_t: how many meters correspond to one tick of incremental encoder
- l: the baseline length (distance between the front wheel and the middle of the rear wheels axis).
- $steer_{off}$: to which angle the orientation of the front wheel corresponds to zero.
- $(pos_{l,x}^r, pos_{l,y}^r, \theta_l^r)$: laser pose w.r.t. robot reference frame.

3 Least Square Method

The least squares method is based on the Gaussian method which iteratively minimizes the function[2]

$$F(\boldsymbol{p}) = \sum_{i=1}^{N} \boldsymbol{e}^{[i]}(\boldsymbol{p})^T \boldsymbol{e}^{[i]}(\boldsymbol{p}) \tag{14}$$

where $\boldsymbol{e}(\boldsymbol{x})$ is the error function or the difference[3] between the $i\!-\!th$ prediction and the $i\!-\!th$ measurement given the current estimate of the parameters configuration p. The goal of this procedure is to find a perturbation $\Delta\boldsymbol{p}$ which minimizes the overall function given N measurements.
The perturbation obtained by minimizing a quadratic approximation of the problem in Δp is

$$F(\boldsymbol{p} + \Delta\boldsymbol{p}) = \Delta\boldsymbol{p}^T \boldsymbol{H}\, \Delta\boldsymbol{p} + 2\, \boldsymbol{b}^T \Delta\boldsymbol{p} + c \tag{15}$$

where

$$\begin{aligned} \boldsymbol{H} &= \sum_{i=1}^{N} \boldsymbol{J}^{[i]\,T} \boldsymbol{J}^{[i]} \\ \boldsymbol{b} &= \sum_{i=1}^{N} \boldsymbol{J}^{[i]\,T} \boldsymbol{e}^{[i]} \\ c &= \sum_{i=1}^{N} \boldsymbol{e}^{[i]\,T} \boldsymbol{e}^{[i]} \\ \boldsymbol{J}^{[i]} &= \frac{\partial \boldsymbol{e}(\boldsymbol{p})}{\partial(\boldsymbol{p})} \quad \text{for} \quad \boldsymbol{p} = \boldsymbol{p}* \end{aligned} \tag{16}$$

[1] time dependence has been omitted for the sake of readability

[2] State and parameters dependency is omitted to simplify the notation

[3] considering that as regards the angles it is not possible to carry out a classical subtraction because they are not found in an eclidean space

Finally, it finds the perturbation $\Delta \boldsymbol{p}$ that cancels the quadratic derivative form

$$\mathbf{0} = \frac{\partial F(\boldsymbol{p} + \Delta \boldsymbol{p})}{\partial \Delta \boldsymbol{p}} \quad \text{or} \quad -\boldsymbol{b} = \boldsymbol{H}\, \Delta \boldsymbol{p} \tag{17}$$

The final pseudo-code is the following one

Algorithm 1 Least Squares Algorithm

$\boldsymbol{H} \leftarrow \mathbf{0}$ $\boldsymbol{b} \leftarrow \mathbf{0}$
for `each measurement i` **do**
  $\boldsymbol{e}^{[i]} \leftarrow \boldsymbol{h}^{[i]}(\boldsymbol{p}*) - \boldsymbol{z}^{[i]}$
  $\boldsymbol{J}^{[i]} \leftarrow \frac{\partial \boldsymbol{e}(\boldsymbol{p})}{\partial (\boldsymbol{p})}$ for $\boldsymbol{p} = \boldsymbol{p}*$
  $\boldsymbol{H} \leftarrow \boldsymbol{H} + \boldsymbol{J}^{[i]\,T} \boldsymbol{J}^{[i]}$
  $\boldsymbol{b} \leftarrow \boldsymbol{b} + \boldsymbol{J}^{[i]\,T} \boldsymbol{e}^{[i]}$
end for
$\Delta \boldsymbol{p} \leftarrow solve\,(-\boldsymbol{b} = \boldsymbol{H}\, \Delta \boldsymbol{p})$
$\boldsymbol{p}* \leftarrow \boldsymbol{p}* + \Delta \boldsymbol{p}$

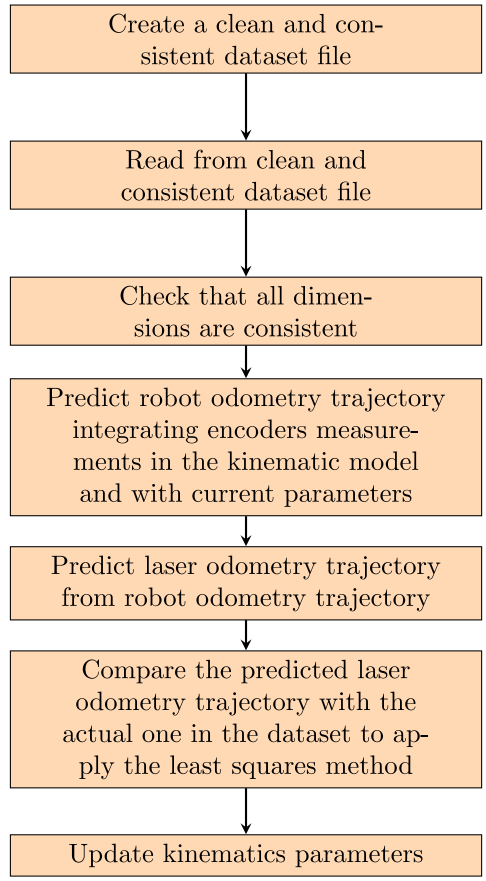
Given the kinematic model complexity and the number of parameters, the Jacobian was measured numerically[4] for each measurement with the following formula

$$\boldsymbol{J}^{[i]}(\boldsymbol{p}*) = \boldsymbol{h}^{[i]}(\boldsymbol{p}* + \Delta \boldsymbol{p}) - \boldsymbol{h}^{[i]}(\boldsymbol{p}* - \Delta \boldsymbol{p}) \tag{18}$$

4 Code

This paragraph presents the main steps to calibrate the kinematic parameters of the robot by having a dataset relating to an odometric trajectory of a laser at a fixed distance with respect to the robot.

[4]this methodology causes a slowdown in the execution of the code but allows the derivative not to be calculated analytically



where the last 4 steps are executed in a loop until convergence is reached, which can be specified as an error threshold or a finite number of rounds.

The first three steps are about the creation, cleaning and examination of data. With *remove unnecessary comments and spaces* function a new dataset is created using the original data and removing comments and unnecessary spaces between words or numbers. This function releases a new dataset text file called *clened dataset*, from which a consistent dataset text file is generated, which takes into account the possible overflow of the incremental encoder variable and starts the timestamp from zero. Starting from the *consistent dataset* file and considering that each row represents a sensor and encoder measurements, the following arrays are created and checked: *timestamp*, *encoders values*, *robot odometry with initial guess* and *laser odometry*. Where

$$\begin{aligned} &encoder\_values[i] = (Abs\_Enc[i], Inc\_Enc[i]) \\ &robot\_odometry\_with\_initial\_guess[i] = (x, y, \theta)[i] \\ &laser\_odometry[i] = (x_L, y_L, \theta_L)[i] \end{aligned} \tag{19}$$

$Abs\_Enc[i]$ is the i-th absolute encoder value, $Inc\_Enc[i]$ is the i-th incremental encoder value, $(x, y, \theta)[i]$ is the i-th robot odometry configuration obtained using initial guess of kinematic parameters and $(x_L, y_L, \theta_L)[i]$ is the i-th sensor, or laser, odometry configuration using calibrated kinematic parameters. Last ones are used as measurements, or ground truth, for the least squares method.

In the *main* file there are two training phases: the first is performed on the entire dataset divided into batches, while in the second the whole dataset is used as a single batch. Several attempts have demonstrated how the first phase is necessary to give the possibility of finding a configuration of the kinematic parameters that would lead to a drastic reduction in the error before finding the configuration such that the error is minimal. If this phase were omitted, the second phase would not reach the minimum error and would cause the error to diverge.

5 Results

In Figure 1 a comparison between the actual and predicted laser odometry trajectory is shown to confirm that the robot is not initially calibrated. The graphs at the top of Figure 1 and 2 show laser position trends w.r.t. its reference system and initial position. While the graphs at the bottom of Figures show the trend of laser orientation. Both predicted tends are achieved using the methodology illustrated in paragraph 2.

By applying the code illustrated in paragraph 4, which exploits the method explained in paragraph 3, a new configuration of the parameters to be calibrated is obtained at each iteration. The final configuration, which involves the minimum error value, determines a predicted laser pose trend similar to the actual one as shown in Figure 2.

The final parameter configuration is as follows

- $k_s = 0.57151$
- $k_t = 0.01107$
- $l = 1.59541$
- $steer_{off} = -0.06878$
- $pos^r_{l,x} = 1.77300$
- $pos^r_{l,y} = 0.01458$
- $\theta^r_l = -5.32804e-04$

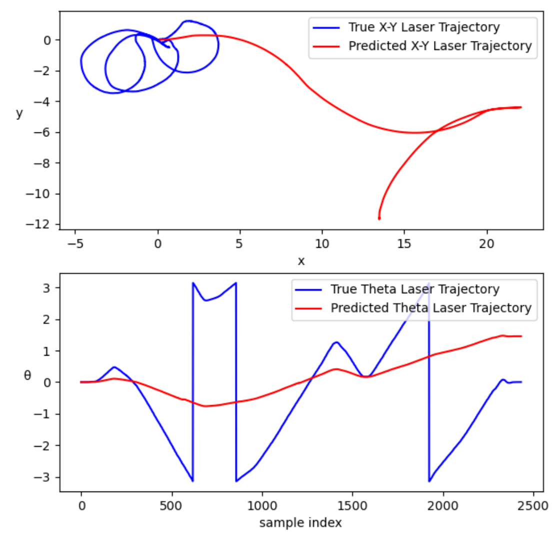


Figure 1: Comparison between true laser odometry (blue line) and predicted laser odometry (red line) with *uncalibrated* parameters (initial guess)

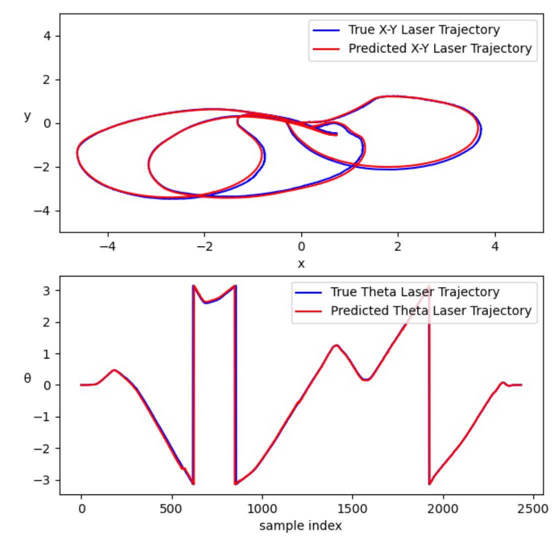


Figure 2: Comparison between true laser odometry (blue line) and predicted laser odometry (red line) with *calibrated* parameters