# Some information about the material for localization with wheel radius identification

The material is provided in the event any of you is interested by going a little beyond what we did in the lab. **Its use and study is not necessary to sucessfully take the lab evaluation and exam. There is absolutely no obligation to study this material.** If you care to use the material and have questions about it or about the result, I will do my best to answer.

The material illustrates a solution which is often used to identify a constant parameter which either is not measurable, or is difficult to measure. It can also be used for parameters which are not strictly constant but evolve slowly over time.

The wheel radius is an example which falls into this category as it can evolve slowly due to wear. For tires, it can also evolve due to changes in tire pressure, which in turn can change over time to due tire temperature.

Of course, the data available (which is the same you used in the lab) does not allow to track slow evolutions of the wheel radius, since the data was recorded over short time intervals. But the idea can be used to identify the wheel radius when it has an initial error. Say you start with the nominal wheel radius and the algorithm identifies the current wheel radius (which depends on wear and tire pressure).

The standard idea is to add the parameter to the state vector. In the example the right and left radii are X(4) and X(5), the first three elements of the state vector being the posture as in the lab.

Of course, you need an evolution model for such a parameter. The simplest way to reflect that the parameter is a constant is to use for parameter $p : p_{k+1} = p_k$. The state equation being disturbed by a state noise, the state noise will allow the parameter to evolve over time. With a very low noise variance, the parameter will be allowed to vary very slowly. To rapidly compensate for a (possibly larger) initial error, a larger variance will be used in the corresponding diagonal terms of $P_{init}$ (4$^{\text{th}}$ and 5$^{\text{th}}$ terms). After the lab, you should already know that a larger initial variance rapidly decreases as a result of taking into account the first measurements.

Since the radii of the wheels are part of the state vector, we cannot use $U = [\Delta D, \Delta\theta]^t$ as input vector. Indeed, that would « hide » the radii and taking the derivative of the evolution function $f$ with respect to the radii would not be possible. So here $U = [\Delta q_r, \Delta q_l]^t$ is used. Note that it would have been possible to make the same choice for the lab without any problem.

Because of this new choice, you will notice that in DefineVariances.m, there is no need for the multiplication by the matrix jointToCartesian.

I have left for you to set the noise variances. Note that if you set the noise of the radii to zero both in $P_{init}$ and $Q\alpha$, then the radii become constant, deterministic variables and you're back to the lab, though with a different definition of the input, but the results should be the same. Of course the covariance matrices are still symmetric positive, but no longer definite (they are not invertible). But the program will run like this if you care to try. But then of course, the KF is not able to compensate for initial errors on the radii…

It's interesting for you to make sure you understand how the evolution model, matrix A, matrix B and matrix C are modified with respect to the lab.