Design of Kalman Filters for Mobile Robots; Evaluation of the Kinematic and Odometric Approach

Thomas Dall Larsen, Karsten Lentfer Hansen, Nils A. Andersen & Ole Ravn
Department of Automation, Technical University of Denmark
Building 326, DK-2800 Lyngby, Denmark
E-mail: tdl@iau.dtu.dk   Fax: +45 45 88 12 95

Keywords: *Mobile robot, modelling, Kalman filter, odometry*

Abstract

Kalman filters have for a long time been widely used on mobile robots as a location estimator. Many different Kalman filter designs have been proposed, using models of various complexity. In this paper, two different design methods are evaluated and compared. Focus is put on the common setup where the mobile robot is equipped with a dual encoder system supported by some additional absolute measurements. A common filter type here, is the odometric *filter where readings from the odometry system on the robot are used together with the geometry of the robot movement as a model of the robot. If additional kinematic assumptions are made, for instance regarding the velocity of the robot, an augmented model can be used instead. This* kinematic *filter has some advantages when used intelligently and it is shown how this type of filter can be used to suppress noise on encoder readings and velocity estimates. The Kalman filter normally consists of a time update followed by one or more data updates. However, it is shown that when using the kinematic filter, the encoder measurements should be fused prior to the time update for better performance.*

1 Introduction

In mobile robot navigation one of the key problems is how to estimate the posture (i.e. the *position* and *pose*) of the robot. By far the most common way of obtaining this estimate is by using a Kalman filter. The Kalman filter uses a system model along with measurements from internal and external sensors to maintain an estimate of the robot's posture and of a corresponding covariance matrix describing the uncertainty of the posture estimate. Knowing the covariance matrices of the estimate and the incoming measurements, the filter fuses measurements and estimate, minimizing the variance of the resulting estimate.

In order for the estimate to remain optimal, it is required that the model of the robot is perfect. As this is rarely the case, most Kalman filters run in a suboptimal and possibly unstable manner. Luckily, the effects of this can be greatly reduced by choosing the filter model intelligently. The most common way in practice to prevent an erroneous filter model to biase or diverge the estimates, is to force the filter to put less confidence in the model and more in the measurements. This is done by increasing the filter's process noise covariance matrix, Q, which is equivalent to adding fictitious process noise in the model to simulate the uncertainties.

As it is impossible to model a real robot perfectly, it is almost always necessary to tune Q when Kalman filters are implemented. As this diminishes not only the influence of the modelling errors but also of the model itself, some considerations should be made regarding the complexity of the model. A thorough and tedious attempt to model the robot followed by a tuning of Q that in practice deteriorates or even discards the outputs from this model, is wasting time both in the design phase and during runtime. Besides, trying to make an accurate dynamical model of the robot contemplating all the nonlinearities caused by for instance friction forces, is not a trivial task and is hardly ever seen in the literature (one example though is found in [1]). The problem (besides the nonlinearities) is that a lot of parameters that change with for instance time and temperature are required to be known quite precisely.

Quite often instead a simple kinematic model assuming either constant velocity or acceleration is used as for instance in [2]. However, as any change in maneuver in these filters is an inherent modelling error, this approach relies heavily on measurements to correct the estimate. Another very common approach, is to use the odometric system of the robot along with a geometric description of the robot movement as the system model as in [3] or [4]. Here, readings from the robot encoders are used, not as measurements, but as inputs driving the filter model. A combination of these filter types can be obtained by augmenting the odometric model with kinematic states as in [5] (for convenience these hybrid filters shall be denoted *kinematic* here).

2 Models for Mobile Robots

As mentioned, mobile robots can be modelled in many ways at many different levels of abstraction. Here, focus is put on *odometric* and *kinematic* models.

2.1 Odometric Kalman Filter

If the mobile robot is equipped with an encoder on each motor shaft, a very feasible and common way of designing the location estimator, is by using these encoder readings as the system model. In this approach, the encoder readings are translated to increases in the mobile robot's translational and rotational position and used as inputs to a simple geometrical filter model. An example of this common type of robot is shown on figure 1. During one sample period the encoders

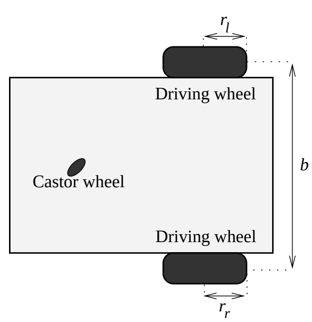

Figure 1: A mobile robot with dual drive and encoders.

will measure angular increments corresponding to the distances d_r and d_l traveled by the right and the left wheel respectively. If the movement of the robot is assumed circular, d_r and d_l can be transformed to a translational and rotational displacement of the robot:

$$\delta d = \frac{d_r + d_l}{2} \quad (1)$$

$$\delta\theta = \frac{d_r - d_l}{b}, \quad (2)$$

where b is the distance between the wheels.

The coordinates of the mobile robot in a global coordinate frame can then be updated by (see [6]):

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \delta d_k \cos(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta d_k \sin(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta\theta_k \end{bmatrix} \quad (3)$$

Equation (3) assumes linear velocity within each sample period. A model assuming circular motion can sometimes be more accurate. This can be obtained by multiplying the X and Y coordinates with an adjustment factor:

$$c = \frac{\sin(\delta\theta_k/2)}{\delta\theta_k/2} \quad (4)$$

Typically the sampling rate is so high compared to the velocity of the robot that the adjustment factor will make very little difference. Equation (3) will therefore be used to describe the robot motion.

The three coordinates (X, Y, θ) constitute the state vector x. In the problem considered in this paper (and frequently encountered in real life) the state is observed by some absolute measurements, z. These measurements are described by a nonlinear function, c, of the robot coordinates and an independent Gaussian noise process, v. Denoting the nonlinear function (3) a, and collecting δd_k and $\delta\theta_k$ in an input vector u_k, the mobile robot can be described by:

$$x_{k+1} = a(x_k, u_k, w_k, k) \quad (5)$$

$$z_k = c(x_k, v_k, k), \quad (6)$$

where: $w_k \sim N(0, q_k), v_k \sim N(0, r_k), E\left[w_i v_j^T\right] = 0$.

An extended Kalman filter can be designed using the system model in equation (3) and the filter equations below as in [3] or [4]:

$$\hat{x}_{k+1} = a(\hat{x}_k, u_k, 0, k) \quad (7)$$

$$P_{k+1} = A_k P_k(+) A_k^T + G_k Q_k G_k^T \quad (8)$$

$$K_k = P_k C_k^T \left[C_k P_k C_k^T + R_k\right]^{-1} \quad (9)$$

$$\hat{x}_k(+) = \hat{x}_k + K_k \left[z_k - C_k \hat{x}_k\right] \quad (10)$$

$$P_k(+) = \left[I - K_k C_k\right] P_k, \quad (11)$$

Denoting $\phi_k = \theta_k + \frac{\delta\theta_k}{2}$, the linearized matrices becomes:

$$A_k = \left.\frac{\partial a}{\partial x_k}\right|_{x_k = \hat{x}_k(+), w_k = 0} = \begin{bmatrix} 1 & 0 & -\delta d_k \sin\phi_k \\ 0 & 1 & \delta d_k \cos\phi_k \\ 0 & 0 & 1 \end{bmatrix}$$

$$C_k = \left.\frac{\partial c}{\partial x_k}\right|_{x_k = \hat{x}_k(+), v_k = 0}$$

$$R_k = \left.\frac{\partial c}{\partial v_k} r_k \frac{\partial c}{\partial v_k}^T\right|_{x_k = \hat{x}_k(+)}$$

$$G_k = \left.\frac{\partial a}{\partial u_k}\right|_{x_k = \hat{x}_k} = \begin{bmatrix} \cos\phi_k & -\frac{1}{2}\delta d \sin\phi_k \\ \sin\phi_k & \frac{1}{2}\delta d \cos\phi_k \\ 0 & 1 \end{bmatrix}$$

The process noise vector, w, is modelled as two independent gaussian white noise processes added to (1) and (2). As shown in [6] this is a fairly good approximation.

2.2 Kinematic Kalman Filter

Assuming that the robot moves at a constant speed, V, and rotates with a constant angular velocity, ω, the state vector can (as in [5]) be augmented to:

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \\ \theta_{k+1} \\ V_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \\ \theta_k \\ V_k \\ \omega_k \end{bmatrix} + \begin{bmatrix} TV_k \cos(\theta_k + \frac{T\omega_k}{2}) \\ TV_k \sin(\theta_k + \frac{T\omega_k}{2}) \\ T\omega_k \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

As the model assumes constant linear and angular velocity, the process noise here is the accelerations: $\dot{V}$ and $\dot{\omega}$. As these are unknown, maximum values $\dot{V}_{max}$ and $\dot{\omega}_{max}$ should be used to ensure that the filter can track the mobile robot.

The encoder readings can now be fused as measurements by dividing (1) and (2) with the sampling time:

$$V_k = \frac{\delta d_{k-1}}{T} \tag{13}$$

$$\omega_k = \frac{\delta \theta_{k-1}}{T} \tag{14}$$

Observe that when the velocities are low or the sampling rate is high, the resolution of (1) and (2) and therefore also (13)-(14) will be poor. This should be taken into account when the process and measurement noise covariance matrices are determined.

Compared with the odometric model the kinematic approach offers a few advantages:

1. The $\dot{V} = \dot{\omega} = 0$ smoothens the encoder measurements and therefore reduces the effects of wheel slippage and limited encoder resolution.
2. Additional measurements of ω or V (from for instance a gyro or a tachometer) can be fused easily. In the odometric filter these have to be fused with δd and $\delta\theta$ before being used as inputs.
3. Provides estimates for the linear and angular velocities (can be used for control).

Of course the velocity estimates could also be obtained simply by using (13)-(14) and an odometric filter, but as shown in [7][1] the Kalman filter estimates can be better than the finite-difference estimates obtained by (13) and (14). (His results indicate, however, that a $\ddot{V} = 0$ model should be used, i.e. (12) should be augmented with the acceleration as well.) The disadvantages of the kinematic approach are:

1. If $\dot{V} \neq 0$ or $\dot{\omega} \neq 0$ the (erroneous) model makes the time update imprecise.
2. The filter is more computationally demanding (not much though).

[1] [7] did not consider a mobile robot but just an encoder shaft. His results, however, can easily be transferred to robots.

The odometric and kinematic filter models are now compared in simulations.

3 Results

To evaluate the performance of the two types of filters, simulations are now performed using an advanced nonlinear Simulink model of the mobile robot contemplating both linear and nonlinear friction forces as well as the dynamics of the robot. The Simulink model is described in [8]. The advantage of using simulations as opposed to physical experiments is that the ground truth is known and the estimation errors therefore can be evaluated. The simulated robot, which like the real robot is equipped with dual encoders sampled every $40ms$ and with a camera detecting guide marks every $3s$, is send down a corridor with visual guide marks placed on both walls as shown in figure 2.

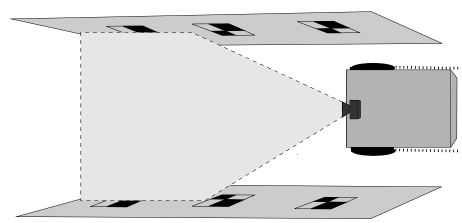

Figure 2: The robot moving down a corridor.

3.1 The Estimation Errors

The performances of the two filters can be observed in figure 3 where the estimation errors for the odometric and kinematic filters are compared. The two filters are seen to perform quite similarly, although the kinematic filter is lagging somewhat when the robot accelerates. Intuitively this is not surprising as the assumption of constant velocity in the filter equations is expected to have some lowpass filtering effect or rather to "fight" changes in velocity reported by the measurements. This, however, is not the real reason for the differences in filter performance. As the kinematic filter is implemented using the standard Kalman filter equations in (7)-(11), the velocity measurements in (13) and (14) are fused *after* the time update in equation (12). As the velocity measurements available at time k in fact are mean velocities for the time period from $k-1$ to k, a more accurate approach would be to fuse the measurements *before* the time update, see figure 4. This will make the time update more accurate as more recent values of the linear and angular velocities are used in the time propagation. A simulation where the data

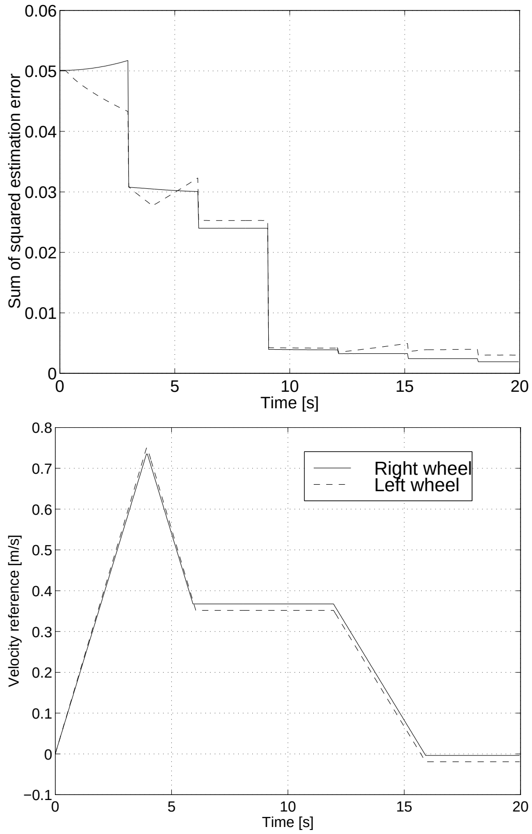

Figure 3: Squared estimation errors for odometric and kinematic (dashed) filter. The lower plot shows the velocity profiles for the driving wheels.

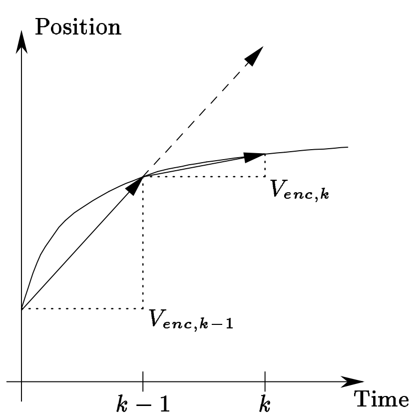

Figure 4: The time update to time k follows the dashed line if it is performed prior to fusing the encoder measurement, $V_{enc,k}$. As $V_{enc,k}$, reports of the mean velocity from time $k-1$ to k, a more accurate time update is obtained by fusing the encoder measurement *before* the time update.

update is performed before the time update, is shown in figure 5. Clearly, the performances of the two filters

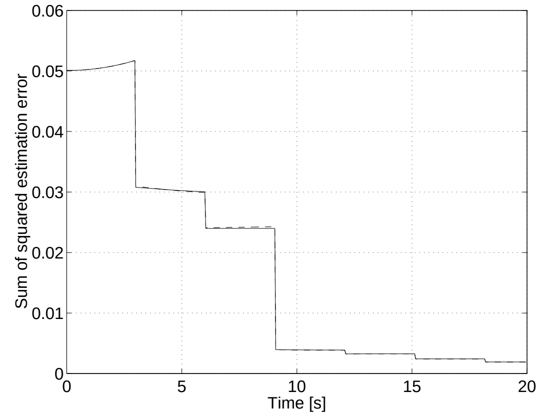

Figure 5: Squared estimation errors for odometric and modified kinematic (dashed) filter.

are more alike here.

3.2 The Velocity Estimates

If the velocity estimates $\hat{V}$ and $\hat{\omega}$ are needed when the odometric filter is used, they must be calculated using equation (13)-(14) which are also denoted the *finite difference* estimates. The estimates from the kinematic filter can be made more smooth and noise rejecting than these, especially if the encoder resolution is relatively low, or the sampling rate is high compared to the velocity of the robot. In figure 6 the velocity estimates using the two methods are compared. It is seen that

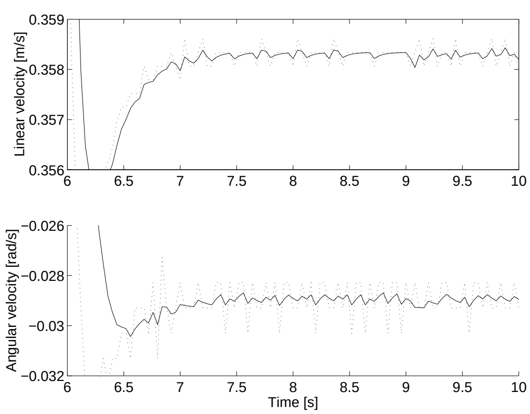

Figure 6: Velocity estimates from kinematic filter and finite difference (dotted).

the estimates from the kinematic filter are less noisy but lag a little. This compromise between noise filtering and fast tracking can be adjusted by tuning the filter noise covariance matrix, Q. When the eigenvalues of Q is increased the kinematic filter will put more weight in the measurements, and in the limit (when $||Q|| \rightarrow \infty$) the velocity estimates from the kinematic filter will equal the finite difference estimates. To illustrate this, two simulations are run with extreme values of Q, see figure 7.

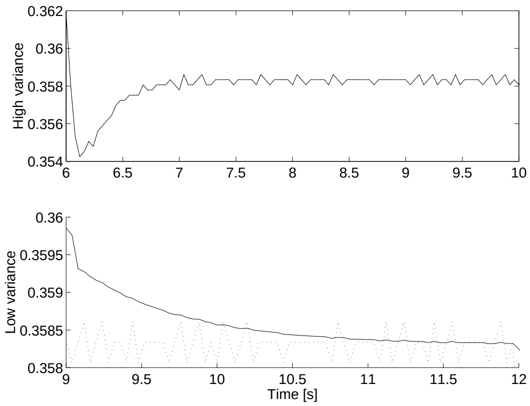

Figure 7: Velocity estimates for kinematic filter and finite difference (dotted). In the uppermost graph the process noise is modelled high and the two methods yield identical results (the two curves are overlapping). In the bottom plot the process noise is modelled low and the estimates from the kinematic filter are seen to be very smooth and slow (lowpass filtered).

In all of the simulations, the encoder resolution was very high and there was no wheel slippage. Under these circumstances, the process noise estimate can therefore be chosen high, and the velocity estimates from the kinematic and odometric filter will be practically identical.

3.3 When the Wheels Slip

The real advantage of the kinematic approach lies in the increased robustness towards erroneous encoder data. This becomes obvious for instance in the presence of wheel slippage. A simulation of this where the left wheel suddenly slips and the encoder readings are erroneous for a period of 2 seconds, is shown in figure 8. Here, the influence of the bad encoder data has been greatly reduced, without affecting the performance of the filter significantly during the usual maneuvers.

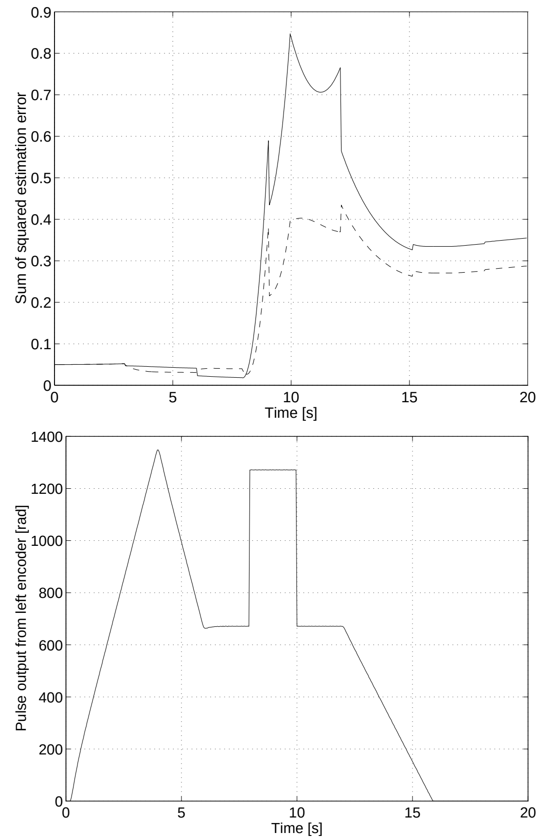

Figure 8: Squared estimation errors for odometric and modified kinematic (dashed) filter when the left wheel slips (from second 8 to 10) as shown on the bottom graph.

4 Conclusion

A comparison between kinematic and odometric filter models for mobile robots was carried out. It was found, that when implementing kinematic filters, it is important to fuse the odometry measurements *prior* to the time update to ensure a truthful interpretation of these. When this is done, it was found that the performance of the two filter types does not differ greatly. However, it was found that the kinematic filter has some advantages that make this filter interesting. Firstly, if the eigenvalues of the process noise matrix are chosen high, the estimates using the kinematic filter will be very similar to the estimates using the odometric filter. Then when the eigenvalues are decreased, the kinematic filter will lowpass filter the estimates and thereby reject noise and make the estimates smoother. Especially when the encoder resolution is low or the wheels slip, this can reduce the estimation error. The kinematic filter therefore provides the designer with one more degree of freedom that can be very useful.

References

[1] Gert L. Andersen, Anders C. Christensen, and Ole Ravn. Augmented models for improving vision control of a mobile robot. In *3rd IEEE Conference on Control Applications*, 1994.

[2] Lindsay Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *Proceedings of the 1992 IEEE*

International Conference on Robotics and Automation, Nice, France, 1996.

[3] Satoshi Murata and Takeshi Hirose. Onboard locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, 40(1), February 1993.

[4] F. Chenavier and J. L. Crowley. Position estimation for a mobile robot using vision and odometry. In *Proceedings of the 1992 IEEE International Conference on Robotics And Automation*, Nice, France, 1992.

[5] Hans-Joachim von der Hardt, Didier Wolf, and René Husson. The dead reckoning localization system of the wheeled mobile robot romane. In *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1996.

[6] C. M. Wang. Location estimation and uncertainty analysis for mobile robots. In *Proceedings of the 1988 International Conference on Robotics and Automation*, 1988.

[7] Pierre R. Bélanger. Estimation of angular velocity and acceleration from shaft encoder measurements. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, 1996.

[8] Magnus Nørgaard, Niels K. Poulsen, and Ole Ravn. The AGV-sim ver. 1.0 - a simulator for autonomous guided vehicles. Technical report, Department of Automation, IAU, 1998.