

# Location Estimation using Delayed Measurements

M. Bak, T. D. Larsen, M. Nørgaard\*, N. A. Andersen, N. K. Poulsen\* & O. Ravn  
Department of Automation, Technical University of Denmark  
Building 326, DK-2800 Lyngby, Denmark

## Abstract

*When combining data from various sensors it is vital to acknowledge possible measurement delays. Furthermore, the sensor fusion algorithm, often a Kalman filter, should be modified in order to handle the delay. This paper examines different possibilities for handling delays and applies a new technique to a sensor fusion system for estimating the location of an Autonomous Guided Vehicle. The system fuses encoder and vision measurements in an extended Kalman filter. Results from experiments in a real environment are reported.*

## 1 Introduction

In a complex control system more than one sensor is often needed to generate and maintain a reliable state estimate. Furthermore, the complexity of the computations in sensors will often introduce significant time delays from the acquisition to the availability of a measurement. This can be due to signal and image processing, data transportation, laboratory tests, etc.

Two examples of systems with delayed measurements are an Autonomous Guided Vehicle (AGV) using vision and a wastewater treatment plant using water samples. Vision is a versatile sensor often used in a location system for detecting natural or artificial guide marks but the recording of an image and the following image processing, that may require extensive computations, will introduce time delays. Faster and more advanced image equipment may decrease the delay but then again more intelligent and complex image algorithms will increase it.

In a wastewater treatment plant the water quality is often monitored by doing comprehensive laboratory tests on water samples resulting in up to several days delayed measurements.

How to incorporate possible unknown time-varying delayed measurements in a state estimation can be non-trivial. As a consequence of sensor-related delays,

a measurement will not reflect the current states of the system it observes and some caution should therefore be taken.

For motion control and navigation it is vital continuously to attain knowledge of location in order to keep a vehicle near its desired path. In AGV applications odometry combined with vision is a widely used approach to location estimation and has proven satisfactory performance [5], [8],[10]. However, the measurement delay handling is often neglected by using either sophisticated hardware, minimal image processing, predefined travelling paths or a *stop & go* strategy, where the AGV stops to process measurements. The latter method has been applied in [2] for docking and navigation through a door. In some cases, though, this is not a feasible strategy and explicit handling of the time delay is necessary.

The purpose of this paper is to focus on the problem of dealing with significant time delays in state estimation using a Kalman filter. This is based on a fast and effective technique described in [7] where delayed measurements easily are incorporated in the state. The method is applied to an AGV using odometry and vision for location estimation where sensor related time delays of 30-40 samples exist in the vision system. The sampling period is 40 ms.

The paper is organized as follows: Section 2 gives a survey of existing techniques for handling the time delays and presents our technique, Section 3 describes the algorithms for estimating the position and orientation of the AGV, section 4 shows the concept through experiments and finally, Section 5 summarizes the results and concludes.

## 2 Delayed Measurements

The presentation herein is based on the Kalman filter equations for a discrete linearized time-varying system with state vector  $x_k$ , input vector  $u_k$  and output vector  $y_k$ . The following lists the equations for notational purpose [6].

---

\*Department of Mathematical Modelling

**System:**

$$x_{k+1} = A_k x_k + B_k u_k + v_k, v_k \in N(0, Q_k) \quad (1)$$

$$y_k = C_k^T x_k + e_k, e_k \in N(0, R_k). \quad (2)$$

**Measurement update:**

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - C_k^T \hat{x}_{k|k-1}) \quad (3)$$

$$P_{k|k} = [I - K_k C_k^T] P_{k|k-1} \quad (4)$$

$$K_k = P_{k|k-1} C_k [C_k^T P_{k|k-1} C_k + R_k]^{-1}. \quad (5)$$

**Time update:**

$$\hat{x}_{k+1|k} = A_k \hat{x}_{k|k} + B_k u_k \quad (6)$$

$$P_{k+1|k} = A_k P_{k|k} A_k^T + Q_k. \quad (7)$$

In case a sensor has a time delay from the acquisition to the availability of a given measurement, the sensor's output equation becomes:

$$y_k = C_{k-d}^T x_{k-d} + e_{k-d}, \quad (8)$$

where  $d$  is the number of delayed samples which is here assumed to be an integer. This means that the measurement is acquired at sampling  $k-d$  but are not available until sampling  $k$ .

## 2.1 Survey of Methods

A number of methods are available for handling the sensor-related time delays such that the measurement reflects current system states rather than prior.

A well-known classical approach, mentioned in e.g. [6], is to augment the state vector in such a way that  $C_{k-d}^T x_{k-d}$  becomes a state. This way, the measurement does reflect a current state and the measurement update can be carried out as usual.

The method is only convenient for constant and small time delays as the state vector is expanded by  $m \times d$  states, where  $m$  is the dimension of the measurement. For time varying and large delays the method is time consuming due to the enlarged state vector and the implementation of the filter is made more difficult provided matrix symmetrical aspects are to be considered in order to reduce the computational burden.

Another computational consuming method is recalculation of the filter upon arrival of a delayed measurement. Providing all measurements, state estimates, and covariance matrices have been stored since the acquisition of the measurement, the filter can be restarted at sampling  $k-d$  and recalculated up to sampling  $k$ . This will obviously stress the filter for large delays since the recalculation preferably should

be carried out in one sample. Further, given a system where a  $d$ -sample delayed measurement arrives at each sampling instance the filtering of a given time interval will be repeated  $d$ -times. This puts great demands on the processing unit.

[1] proposes a method for how to minimize the required computations in recalculating the filter by updating the covariance matrix  $P_k$  and the state  $\hat{x}_k$  at different times. Given the covariance of the measurement,  $R_k$  and the measurement matrix,  $C_k^T$  are known at the time of acquisition,  $k-d$ , the covariance of the state,  $P_k$ , can be updated at sampling  $k-d$  and at sampling  $k$ , a correction to the state can quite easily be made calculated and applied. The disadvantage of this method is that in the time interval  $k-d$  to  $k$  the accuracy of the state is wrong (the  $P$  matrix has wrongly been updated) and this can lead to misvalidation of other measurements arriving in the time interval.

In [7] an extension to this method is proposed that avoids the problem of wrongly updating the covariance matrix by having two filters running in parallel. One filter runs with and one filter runs optimal without the updated covariance matrix  $P_K$ .

## 2.2 Interpolating Measurements

As mentioned in section 2.1 the method for handling sensor-related delays by [1] requires the knowledge of the measurement matrix,  $C_{k-d}^T$  and the covariance matrix  $R_{k-d}$  at the time of the acquisition. This might not be the case as with our AGV used for the experiments where the covariance on the vision measurements very much depend on the distance and the angle to the observed object, i.e. the uncertainty on the distance to an object is proportional with the actual distance. In order to avoid the computational expensive recalculation of the filter, a new method is needed. The following method is described in detail in [7] and shall here be adopted for our application.

Basically, the idea is to interpolate the measurement so that it reflects current states rather than prior. Heuristically, this is done by modifying the measurement based on the change in the system state in the time interval from acquisition to availability. This leads to the following interpolated measurement  $y_k^i$ :

$$y_k^i = y_{k-d} + C_k^T \hat{x}_k - C_{k-d}^T \hat{x}_{k-d}. \quad (9)$$

As a consequence of the adjustment, the Kalman Gain in the measurement update should be decreased to signal less belief in the incorporated measurement. [7] lists the following equation for the gain in case only

delayed measurements are fused:

$$K_k^i = \begin{bmatrix} 1 & 0 & \sum_{j=k-d}^k A_{13,j} \\ 0 & 1 & \sum_{j=k-d}^k A_{23,j} \\ 0 & 0 & 1 \end{bmatrix} \times K_k \quad (10)$$

Obviously, this is a simple computational cheap method.

### 3 Location Estimation for an AGV

The following describes the location system on-board the AGV.

#### 3.1 Odometry

The two driving wheels are equipped with encoders and at each sampling instance the encoder pulses provides information about the rotation of the wheel and thereby of the distance  $\tau_r$  and  $\tau_l$  driven by the right and the left wheel respectively. The reference point on the AGV is given by the mid point of the two wheels' axis and this point's displacement  $\Delta S$  and rotational change  $\Delta\Theta$  can be determined by:

$$\Delta S = \frac{\tau_r + \tau_l}{2} \quad (11)$$

$$\Delta\Theta = \frac{\tau_r - \tau_l}{B}, \quad (12)$$

where  $B$  is the distance between the two wheels.

The position and orientation of the AGV in a two-dimensional space can at sampling instance  $k$  be represented by the triplet  $[X_k \ Y_k \ \Theta_k]^T$  where  $X_k, Y_k$  is the position and  $\Theta_k$  is the orientation or heading of the robot. The AGV's position and rotation at sampling instance  $k+1$  can be determined by the well-known odometry model (see [11]), providing the displacement and rotational change:

$$X_{k+1} = X_k + \Delta S \frac{\sin(\frac{\Delta\Theta}{2})}{\frac{\Delta\Theta}{2}} \cos(\Theta_k + \frac{\Delta\Theta}{2}) \quad (13)$$

$$Y_{k+1} = Y_k + \Delta S \frac{\sin(\frac{\Delta\Theta}{2})}{\frac{\Delta\Theta}{2}} \sin(\Theta_k + \frac{\Delta\Theta}{2}) \quad (14)$$

$$\Theta_{k+1} = \Theta_k + \Delta\Theta. \quad (15)$$

This model assumes the speed of the robot is constant between two sampling instances.

By introducing a state vector  $x_k$  and an input vector  $u_k$  given as:

$$x_k = \begin{bmatrix} X_k \\ Y_k \\ \Theta_k \end{bmatrix} \quad u_k = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, \quad (16)$$

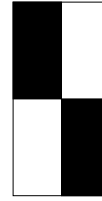


Figure 1: Guide mark used by vision.

and assuming that any noise  $v_k$  attacking the system enters through  $u_k$ , the equations (13)-(15) can be rewritten as:

$$x_{k+1} = f(x_k, u_k + v_k), \quad (17)$$

where  $f$  is nonlinear and given by (13)-(15).

The noise  $v_k$  comes from a number of sources such as uneven floors, wheel slippage, finite encoder resolution, misdetermined geometrical quantities (wheel radius, wheel distance), etc. This is further described in details in [4].

#### 3.2 Vision

The vision system on-board the AGV uses a 2-dimensional guide mark of the type shown in Fig. 1 for location estimation. The dimension of the mark is  $0.24 \times 0.1$  m. An image of a guide mark provides a measurement of the full state  $x_k$  but due to image recording and especially processing it will be delayed of 70-75 samples. This time delay is time-varying depending on the size and position of the guide mark and the number of potential guide marks in the image. Furthermore, it will be contaminated by some noise  $e_k$  mainly due to the discretation into pixels in the camera. This can be described by the following equation

$$y_k = x_{k-d} + e_{k-d}, \quad (18)$$

where  $d$  is the number of delayed samples.

#### 3.3 Fusion of Data with EKF

Equations (17) and (18) describe our location system and an extended Kalman filter (EKF) can now be applied for estimating the position and orientation by fusing the information from the encoders and the camera.

Given encoder measurements, the state estimate  $\hat{x}_k$  can be updated using (17):

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k), \quad (19)$$

and the covariance of the state  $P_k$  is updated using a linearization of (17):

$$P_{k+1|k} = F_k P_{k|k} F_k^T + G_k Q_k G_k^T, \quad (20)$$

where  $F_k$  and  $G_k$  are Jacobian matrices of  $f_k$  with respect to  $\hat{x}_k$  and  $u_k$  respectively.  $Q_k$  is the covariance of the process noise  $v_k$ .

The noise sequences  $\{v_k\}$  and  $\{e_k\}$  are assumed to be white and Gaussian with the following properties:

$$v_k \in N(0, Q_k); \quad \text{Cov}(v_i, v_j) = Q_k \delta_{i-j} \quad (21)$$

$$e_k \in N(0, R_k); \quad \text{Cov}(e_i, e_j) = R_k \delta_{i-j}, \quad (22)$$

where  $\delta_k$  is Kronecker's delta function. Furthermore,  $v_k$  and  $e_k$  are assumed mutually independent, so that  $\text{Cov}(v_i, e_j) = 0$  for all  $i, j$ .

A delayed measurement is interpolated as described in section 2.2 and updated as in a standard Kalman filter (equations (3)-(5)) with measurement matrix  $C_k = I$ . As the covariance of a vision measurement is modelled as a diagonal matrix, this implies that the components of the measurement  $X, Y$  and  $\Theta$  are independent and accordingly can be split up in three measurements which can be processed one by one. This is advantageous regarding computational aspects and simplifies the implementation of a numerical stable algorithm which often builds on scalar measurements [3].

The vision system is limited in detection of guide marks due to a limited field of view for the camera and a constrained number of guide marks which implies interrupts in the measurements. Thus, the following algorithm has been applied.

**No vision measurements.** Only the time update is executed:  $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k}$ .

**One vision measurement.** Interpolate the measurement and its covariance and execute both the time update and the measurement update.

**Additional vision measurements.** As more than one guide mark may be in the field of view more than one vision measurement may arrive at the same time. Here, the measurement update is executed once for each measurement.

## 4 Experiments

Experiments were carried out in the laboratory in order to examine and verify the location system. The test bed AGV is described in more details in [9].

### 4.1 Experimental Setup

The AGV drive system consists of two DC motors mounted on the front wheels. They are equipped with encoders with an resolution of 800 pulses/turn. A standard commercial  $512 \times 512$  pixel grey level type CCD camera is mounted on top of the AGV with an on-board CPU for image processing. The control and coordination of the AGV is handled by an on-board MVME-162A main board with a MC 68040 CPU running the OS-9 operating system.

In order to evaluate the performance of the location system it is crucial to know the exact position and orientation of the AGV during an experiment. Here this is attained through a surveillance camera placed in the ceiling of the laboratory.

The AGV is equipped with two light diodes, one on each side of the AGV, which are tracked with the surveillance camera. The position and orientation of the AGV can then be monitored with suitable image processing and knowledge of the camera's internal parameters (such as focal length, center pixels, etc.) and the translation and rotation of the camera relative to some global reference coordinate system. This can be achieved using calibration tiles with known geometry.

The recording of an image is synchronized with the AGV which allows for conformity in the ground truth and local AGV data. The uncertainty on the ground truth is estimated to be under 10 mm in position and  $0.15^\circ$  in orientation. Due to a limited field of view, this tracking system allows for tracking the AGV over an area of  $1.75 \text{ m} \times 1.75 \text{ m}$ . Additional cameras can easily be applied to enlarge the tracking region but makes the calibration task somewhat more difficult.

The following scenario is used for evaluation. The AGV travels parallel to a wall on which a guide mark is placed. The camera is panned in order to view the guide mark but otherwise kept in a constant position. The speed of the AGV is held under 0.15 m/s. Fig 2 shows the AGV trajectory. Before commencing the ex-

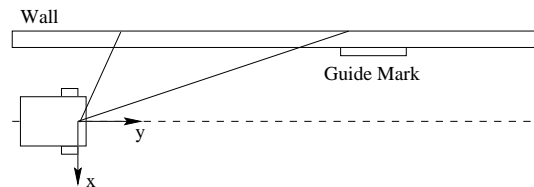


Figure 2: The AGV trajectory.

periments, the systematical odometric errors has successfully been minimized by calibrating the distance between the wheels and the wheel diameters. This was

carried out using the method in [4] and has resulted in a mean value positional drift of 2.8mm/m and orientational drift of  $0.003^\circ/m$  attained on an even floor without door steps or the like, which must be considered an unnatural working environment. The realistic drift is therefore higher.

The parameters in the extended Kalman filter have been selected through calculations and experiments/observations. The stochastic errors effecting the odometry has been modelled in the covariance matrix  $Q$  and is selected as follows:

$$Q = \begin{bmatrix} (0.1 \text{ mm})^2 & 0 \\ 0 & (0.1 \text{ mm})^2 \end{bmatrix}. \quad (23)$$

The covariance matrix,  $R_k$ , for the vision measurements has been found by linearizing the set of equations used to transform the pixel-coordinates provided by the vision system into the AGV's position and orientation. Two temporary variables are used to determine the uncertainty: the distance  $d_{gm}$  from a guide mark to the AGV and the angle  $\Theta_{gm}$  between the normal to a guide mark and the line-of-sight of the AGV camera. The uncertainty on the distance  $d_{gm}$  is found to depend solely on the actual distance as shown in Fig. 3 while the uncertainty on the angle  $\Theta_{gm}$  is found to depend on both the distance and the actual angle as shown in Fig. 4. Knowing the uncertainty

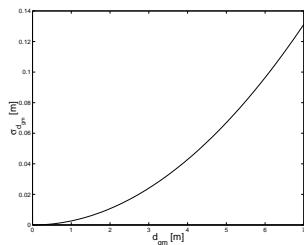


Figure 3: Standard deviation for  $d_{gm}$ .

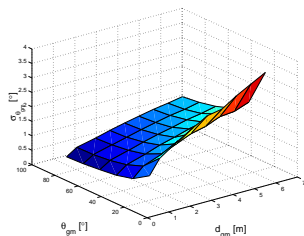


Figure 4: Standard deviation for  $\Theta_{gm}$ .

on these two quantities, the covariance matrix  $R_k$  can be estimated.

Finally, the initial values of the kalman filter (the state and the state error covariance matrix) have been selected as follows, assembling a realistic starting point:

$$x - \hat{x}_{init} = \begin{bmatrix} 0.01m \\ 0.01m \\ 2.00^\circ \end{bmatrix} \quad (24)$$

$$P_{init} = (0.01)^2 \times I. \quad (25)$$

## 4.2 Results

Fig. 5 and 6 show the location results from the first run where the delay is handled with the interpolation method. Fig. 7 and 8 show the results where the delayed measurements are incorporated as they are. The effect from the time delay handling is only significant in the  $y$  state, which is the moving direction of the AGV. In case the heading of the AGV is changed during the time delay period, the effect is expected to be much more crucial.

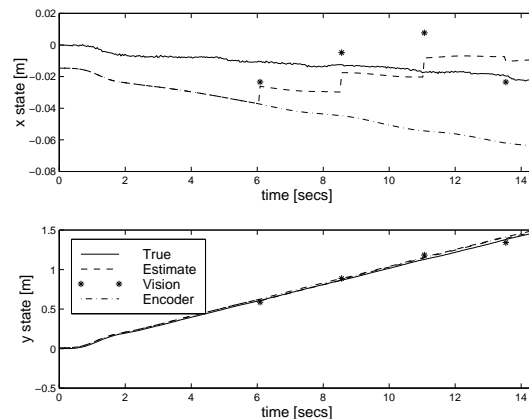


Figure 5: X and Y estimates with time delay handling.

## 5 Conclusions

In this paper different ways of approaching delayed measurements in a Kalman filter have been described. An algorithm for interpolating a delayed measurement minimizes the computational burden even for significant time delays. On our AGV, we have successfully applied this algorithm for dealing with delayed vision measurements in a locating system. By using an extended Kalman filter for fusing data from encoders and camera, the location estimate shows satisfactory accuracy.

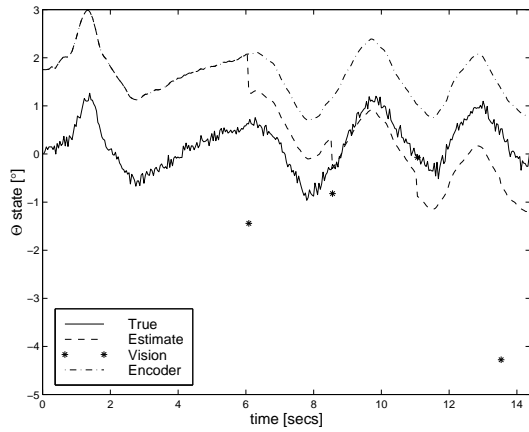


Figure 6:  $\Theta$  estimate with time delay handling.

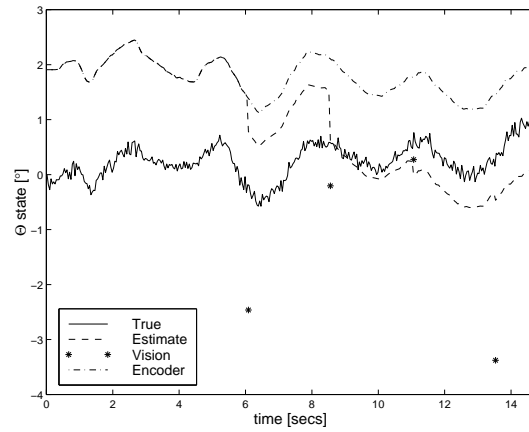


Figure 8:  $\Theta$  estimate without time delay handling.

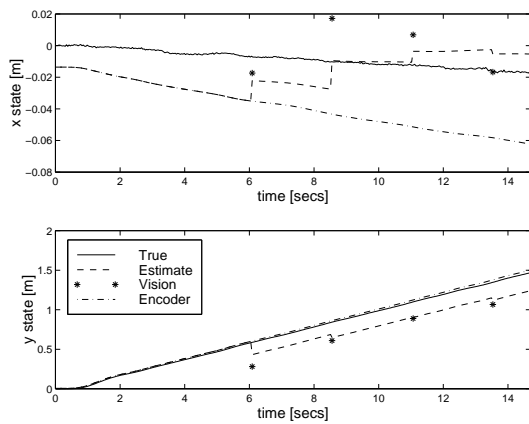


Figure 7:  $X$  and  $Y$  estimates without time delay handling.

## References

- [1] Harold L. Alexander. State estimation for distributed systems with sensing delay. *SPIE*, 1470, 1991.
- [2] Nils Andersen, Lars Henriksen, and Ole Ravn. Design of navigation and control for an agv. In *Proceedings of the 2. IFAC Conference on Intelligent Autonomous Vehicles*, Helsinki, 1995.
- [3] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press, 1977.
- [4] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, 12(6), December 1996.
- [5] 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.
- [6] Arthur Gelb. *Applied Optimal Estimation*. The MIT Press, 1974.
- [7] Thomas D. Larsen, Niels K. Poulsen, Nils A. Andersen, and Ole Ravn. Incorporation of time delayed measurements in a discrete-time kalman filter. To appear, 1998.
- [8] 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.
- [9] Ole Ravn and Nils A. Andersen. A test bed for experiments with intelligent vehicles. In *Proceedings of the First Workshop on Intelligent Autonomous Vehicles*, Southampton, England, 1995.
- [10] E. Stella, G. Cirelli, F. P. Lovergine, and A. Distanto. Position estimation for a mobile robot using data fusion. In *Proceedings of the 1995 IEEE International Symposium on Intelligent Control*, Monterey, CA, USA, 1995.
- [11] C. Ming Wang. Location estimation and uncertainty analysis for mobile robots. In *Proceedings of the 1988 International Conference on Robotics and Automation*, Philadelphia, 1988.