density function. So the resulting 1000 locations will be concentrated primarily at the highest probability locations. This biasing is desirable, but only to a point.

We also wish to ensure that *some* less likely locations are tracked, as otherwise, if the robot does indeed receive unlikely sensor measurements, it will fail to localize. This *randomization* of the sampling process can be performed by adding additional samples from a flat distribution, for example. Further enhancements of these randomized methods enable the number of statistical samples to be varied on the fly, based, for instance, on the ongoing localization confidence of the system. This further reduces the number of samples required on average while guaranteeing that a large number of samples will be used when necessary [68].

These sampling techniques have resulted in robots that function indistinguishably as compared to their full belief state set ancestors, yet use computationally a fraction of the resources. Of course, such sampling has a penalty: completeness. The probabilistically complete nature of Markov localization is violated by these sampling approaches because the robot is failing to update *all* the nonzero probability locations, and thus there is a danger that the robot, due to an unlikely but correct sensor reading, could become truly lost. Of course, recovery from a lost state is feasible just as with all Markov localization techniques.

5.6.3 Kalman filter localization

The Markov localization model can represent any probability density function over robot position. This approach is very general but, due to its generality, inefficient. Consider instead the key demands on a robot localization system. One can argue that it is not the exact replication of a probability density curve but the *sensor fusion* problem that is key to robust localization. Robots usually include a large number of heterogeneous sensors, each providing clues as to robot position and, critically, each suffering from its own failure modes. Optimal localization should take into account the information provided by all of these sensors. In this section we describe a powerful technique for achieving this sensor fusion, called the Kalman filter. This mechanism is in fact more efficient than Markov localization because of key simplifications when representing the probability density function of the robot's belief state and even its individual sensor readings, as described below. But the benefit of this simplification is a resulting *optimal recursive data-processing algorithm*. It incorporates all information, regardless of precision, to estimate the current value of the variable of interest (i.e., the robot's position). A general introduction to Kalman filters can be found in [106] and a more detailed treatment is presented in [3].

Figure 5.25 depicts the general scheme of Kalman filter estimation, where a system has a control signal and system error sources as inputs. A measuring device enables measuring some system states with errors. The Kalman filter is a mathematical mechanism for producing an optimal estimate of the system state based on the knowledge of the *system* and the *measuring device*, the description of the system noise and measurement errors and the

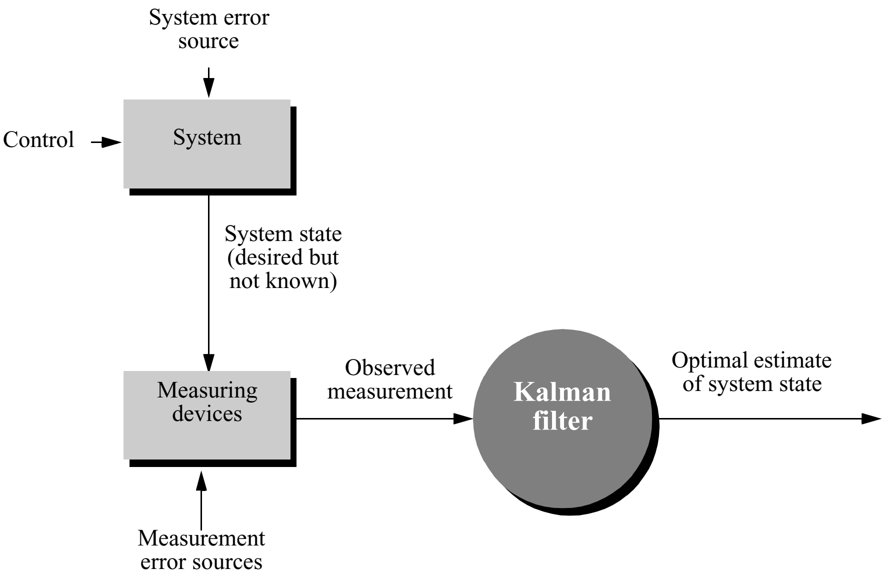

Figure 5.25
Typical Kalman filter application [106].

uncertainty in the dynamics models. Thus the Kalman filter *fuses* sensor signals and system knowledge in an optimal way. Optimality depends on the criteria chosen to evaluate the performance and on the assumptions. Within the Kalman filter theory the system is assumed to be *linear* and *white* with *Gaussian* noise. As we have discussed earlier, the assumption of Gaussian error is invalid for our mobile robot applications but, nevertheless, the results are extremely useful. In other engineering disciplines, the Gaussian error assumption has in some cases been shown to be quite accurate [106].

We begin with a section that introduces Kalman filter theory, then we present an application of that theory to the problem of mobile robot localization (5.6.3.2). Finally, section 5.6.3.2 presents a case study of a mobile robot that navigates indoor spaces by virtue of Kalman filter localization.

5.6.3.1 Introduction to Kalman filter theory

The basic Kalman filter method allows multiple measurements to be incorporated optimally into a single estimate of state. In demonstrating this, first we make the simplifying assumption that the state does not change (e.g., the robot does not move) between the acquisition of the first and second measurement. After presenting this static case, we can introduce *dynamic prediction* readily.

Static estimation. Suppose that our robot has two sensors, an ultrasonic range sensor and a laser rangefinding sensor. The laser rangefinder provides far richer and more accurate data for localization, but it will suffer from failure modes that differ from that of the sonar ranger. For instance, a glass wall will be transparent to the laser but, when measured head-on, the sonar will provide an accurate reading. Thus we wish to combine the information provided by the two sensors, recognizing that such sensor fusion, when done in a principled way, can only result in information gain.

The Kalman filter enables such fusion extremely efficiently, as long as we are willing to approximate the error characteristics of these sensors with unimodal, zero-mean, Gaussian noise. Specifically, assume we have taken two measurements, one with the sonar sensor at time k and one with the laser rangefinder at time $k+1$. Based on each measurement individually we can estimate the robot's position. Such an estimate derived from the sonar is q_1 and the estimate of position based on the laser is q_2. As a simplified way of characterizing the error associated with each of these estimates, we presume a (unimodal) Gaussian probability density curve and thereby associate one variance with each measurement: σ_1^2 and σ_2^2. The two dashed probability densities in figure 5.26 depict two such measurements. In summary, this yields two robot position estimates:

$$\hat{q}_1 = q_1 \text{ with variance } \sigma_1^2 \tag{5.28}$$

$$\hat{q}_2 = q_2 \text{ with variance } \sigma_2^2 . \tag{5.29}$$

The question is, how do we *fuse* (combine) these data to get the best estimate $\hat{q}$ for the robot position? We are assuming that there was no robot motion between time k and time $k+1$, and therefore we can directly apply the same weighted least-squares technique of equation (5.26) in section 4.3.1.1. Thus we write

$$S = \sum_{i=1}^{n} w_i(\hat{q} - q_i)^2 \tag{5.30}$$

with w_i being the weight of measurement i. To find the minimum error we set the derivative of S equal to zero.

$$\frac{\partial S}{\partial \hat{q}} = \frac{\partial}{\partial \hat{q}} \sum_{i=1}^{n} w_i(\hat{q} - q_i)^2 = 2\sum_{i=1}^{n} w_i(\hat{q} - q_i) = 0 \tag{5.31}$$

$$\sum_{i=1}^{n} w_i \hat{q} - \sum_{i=1}^{n} w_i q_i = 0 \tag{5.32}$$

$$\hat{q} = \frac{\sum_{i=1}^{n} w_i q_i}{\sum_{i=1}^{n} w_i} \tag{5.33}$$

If we take as the weight w_i

$$w_i = \frac{1}{\sigma_i^2} \tag{5.34}$$

then the value of $\hat{q}$ in terms of two measurements can be defined as follows:

$$\hat{q} = \frac{\frac{1}{\sigma_1^2} q_1 + \frac{1}{\sigma_2^2} q_2}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} q_2 \tag{5.35}$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = \frac{\sigma_2^2 + \sigma_1^2}{\sigma_1^2 \sigma_2^2} \quad ; \quad \sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} \tag{5.36}$$

Note that from equation (5.36) we can see that the resulting variance σ^2 is less than all the variances σ_i^2 of the individual measurements. Thus the uncertainty of the position estimate has been decreased by combining the two measurements. The solid probability density curve represents the result of the Kalman filter in figure 5.26, depicting this result. Even poor measurements, such as are provided by the sonar, will only increase the precision of an estimate. This is a result that we expect based on information theory.

Equation (5.35) can be rewritten as

$$\hat{q} = q_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}(q_2 - q_1) \tag{5.37}$$

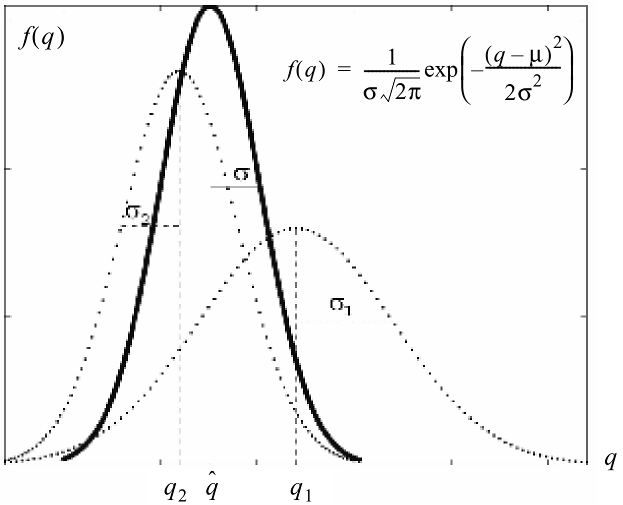

Figure 5.26
Fusing probability density of two estimates [106].

or, in the final form that is used in Kalman filter implementation,

$$\hat{x}_{k+1} = \hat{x}_k + K_{k+1}(z_{k+1} - \hat{x}_k) \tag{5.38}$$

where

$$K_{k+1} = \frac{\sigma_k^2}{\sigma_k^2 + \sigma_z^2} \;;\; \sigma_k^2 = \sigma_1^2 \;;\; \sigma_z^2 = \sigma_2^2 \tag{5.39}$$

Equation (5.38) tells us, that the best estimate $\hat{x}_{k+1}$ of the state x_{k+1} at time $k+1$ is equal to the best prediction of the value $\hat{x}_k$ before the new measurement z_{k+1} is taken, plus a correction term of an optimal weighting value times the difference between z_{k+1} and the best prediction $\hat{x}_k$ at time $k+1$. The updated variance of the state $\hat{x}_{k+1}$ is given using equation (5.36)

$$\sigma_{k+1}^2 = \sigma_k^2 - K_{k+1}\sigma_k^2 \tag{5.40}$$

The new, fused estimate of robot position provided by the Kalman filter is again subject to a Gaussian probability density curve. Its mean and variance are simply functions of the inputs' means and variances. Thus the Kalman filter provides both a compact, simplified representation of uncertainty and an extremely efficient technique for combining heterogeneous estimates to yield a new estimate for our robot's position.

Dynamic estimation. Next, consider a robot that moves between successive sensor measurements. Suppose that the motion of the robot between times k and $k+1$ is described by the velocity u and the noise w which represents the uncertainty of the actual velocity:

$$\frac{dx}{dt} = u + w \tag{5.41}$$

If we now start at time k, knowing the variance σ_k^2 of the robot position at this time and knowing the variance σ_w^2 of the motion, we obtain for the time k' just when the measurement is taken,

$$\hat{x}_{k'} = \hat{x}_k + u(t_{k+1} - t_k) \tag{5.42}$$

$$\sigma_{k'}^2 = \sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] \tag{5.43}$$

where

$t_{k'} = t_{k+1}$;

t_{k+1} and t_k are the time in seconds at $k+1$ and k respectively.

Thus $\hat{x}_{k'}$ is the optimal prediction of the robot's position just as the measurement is taken at time $k+1$. It describes the growth of position error until a new measurement is taken (figure 5.27).

We can now rewrite equations (5.38) and (5.39) using equations (5.42) and (5.43).

$$\begin{aligned}\hat{x}_{k+1} &= \hat{x}_{k'} + K_{k+1}(z_{k+1} - \hat{x}_{k'}) \\ &= [\hat{x}_k + u(t_{k+1} - t_k)] + K_{k+1}[z_{k+1} - \hat{x}_k - u(t_{k+1} - t_k)]\end{aligned} \tag{5.44}$$

$$K_{k+1} = \frac{\sigma_{k'}^2}{\sigma_{k'}^2 + \sigma_z^2} = \frac{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k]}{\sigma_k^2 + \sigma_w^2[t_{k+1} - t_k] + \sigma_z^2} \tag{5.45}$$

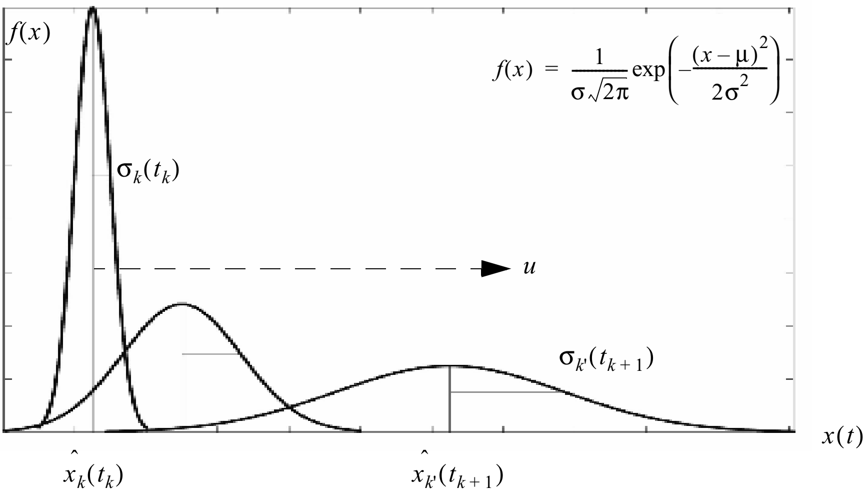

Figure 5.27
Propagation of probability density of a moving robot [106].

The optimal estimate at time $k+1$ is given by the last estimate at k and the estimate of the robot motion including the estimated movement errors.

By extending the above equations to the vector case and allowing time-varying parameters in the system and a description of noise, we can derive the Kalman filter localization algorithm.

5.6.3.2 Application to mobile robots: Kalman filter localization

The Kalman filter is an optimal and efficient sensor fusion technique. Application of the Kalman filter to localization requires posing the robot localization problem as a sensor fusion problem. Recall that the basic probabilistic update of robot belief state can be segmented into two phases, *perception update* and *action update,* as specified by equations (5.21) and (5.22).

The key difference between the Kalman filter approach and our earlier Markov localization approach lies in the perception update process. In Markov localization, the entire perception, that is, the robot's set of instantaneous sensor measurements, is used to update each possible robot position in the belief state individually using the Bayes formula. In some cases, the perception is abstract, having been produced by a feature extraction mechanism, as in Dervish. In other cases, as with Rhino, the perception consists of raw sensor readings.

By contrast, perception update using a Kalman filter is a multistep process. The robot's total sensory input is treated not as a monolithic whole but as a set of extracted features that