$$
\begin{align*}
\hat{p}(k+1 \mid k+1) & =\hat{p}(k+1 \mid k)+K(k+1) \cdot v(k+1) \\
& =\hat{p}(k+1 \mid k)+K(k+1) \cdot\left[z_{j}(k+1)-h_{i}\left(z_{t}, \hat{p}(k+1 \mid k)\right)\right] \\
& =\hat{p}(k+1 \mid k)+K(k+1) \cdot\left[z_{j}(k+1)-z_{t}\right] \tag{5.57}
\end{align*}
$$

corresponding to equation (5.44).

### 5.6.3.3 Case study: Kalman filter localization with line feature extraction

The Pygmalion robot at EPFL is a differential-drive robot that uses a laser rangefinder as its primary sensor [37, 38]. In contrast to both Dervish and Rhino, the environmental representation of Pygmalion is continuous and abstract: the map consists of a set of infinite lines describing the environment. Pygmalion's belief state is, of course, represented as a Gaussian distribution since this robot uses the Kalman filter localization algorithm. The value of its mean position $\mu$ is represented to a high level of precision, enabling Pygmalion to localize with very high precision when desired. Below, we present details for Pygmalion's implementation of the five Kalman filter localization steps. For simplicity we assume that the sensor frame $\{S\}$ is equal to the robot frame $\{R\}$. If not specified all the vectors are represented in the world coordinate system $\{W\}$.

1. Robot position prediction. At the time increment $k$ the robot is at position $p(k)=[x(k) y(k) \theta(k)]^{T}$ and its best position estimate is $\hat{p}(k \mid k)$. The control input $u(k)$ drives the robot to the position $p(k+1)$ (figure 5.29).

The robot position prediction $\hat{p}(k+1)$ at the time increment $k+1$ can be computed from the previous estimate $\hat{p}(k \mid k)$ and the odometric integration of the movement. For the differential drive that Pygmalion has we can use the model (odometry) developed in section 5.2.4:

$$
\hat{p}(k+1 \mid k)=\hat{p}(k \mid k)+u(k)=\hat{p}(k \mid k)+\left[\begin{array}{c}
\frac{\Delta s_{r}+\Delta s_{l}}{2} \cos \left(\theta+\frac{\Delta s_{r}-\Delta s_{l}}{2 b}\right)  \tag{5.58}\\
\frac{\Delta s_{r}+\Delta s_{l}}{2} \sin \left(\theta+\frac{\Delta s_{r}-\Delta s_{l}}{2 b}\right) \\
\frac{\Delta s_{r}-\Delta s_{l}}{b}
\end{array}\right]
$$

with the updated covariance matrix


Figure 5.29
Prediction of the robot's position (thick) based on its former position (thin) and the executed movement. The ellipses drawn around the robot positions represent the uncertainties in the $x, y$ direction (e.g.; $3 \sigma$ ). The uncertainty of the orientation $\theta$ is not represented in the picture.

$$
\begin{equation*}
\Sigma_{p}(k+1 \mid k)=\nabla_{p} f \cdot \Sigma_{p}(k \mid k) \cdot \nabla_{p} f^{T}+\nabla_{u} f \cdot \Sigma_{u}(k) \cdot \nabla_{u} f^{T} \tag{5.59}
\end{equation*}
$$

where

$$
\Sigma_{u}=\operatorname{cov}\left(\Delta s_{r}, \Delta s_{l}\right)=\left[\begin{array}{cc}
k_{r}\left|\Delta s_{r}\right| & 0  \tag{5.60}\\
0 & k_{l}\left|\Delta s_{l}\right|
\end{array}\right]
$$

2. Observation. For line-based localization, each single observation (i.e., a line feature) is extracted from the raw laser rangefinder data and consists of the two line parameters $\beta_{0, j}$, $\beta_{1, j}$ or $\alpha_{j}, r_{j}$ (figure 4.36) respectively. For a rotating laser rangefinder, a representation in the polar coordinate frame is more appropriate and so we use this coordinate frame here:

$$
z_{j}(k+1)={ }^{R}\left[\begin{array}{l}
\alpha_{j}  \tag{5.61}\\
r_{j}
\end{array}\right]
$$

After acquiring the raw data at time $k+1$, lines and their uncertainties are extracted (figure $5.30 \mathrm{a}, \mathrm{b}$ ). This leads to $n_{0}$ observed lines with $2 n_{0}$ line parameters (figure 5.30 c ) and a covariance matrix for each line that can be calculated from the uncertainties of all the
measurement points contributing to each line as developed for line extraction in section 4.3.1.1:

$$
\Sigma_{R, j}=\left[\begin{array}{cc}
\sigma_{\alpha \alpha} & \sigma_{\alpha r}  \tag{5.62}\\
\sigma_{r \alpha} & \sigma_{r r}
\end{array}\right]_{j}
$$



Figure 5.30
Observation: From the raw data (a) acquired by the laser scanner at time $k+1$, lines are extracted (b). The line parameters $\alpha_{j}$ and $r_{j}$ and its uncertainties can be represented in the model space (c).
3. Measurement prediction. Based on the stored map and the predicted robot position $\hat{p}(k \mid k)$, the measurement predictions of expected features $z_{t, i}$ are generated (figure 5.31). To reduce the required calculation power, there is often an additional step that first selects the possible features, in this case lines, from the whole set of features in the map. These lines are stored in the map and specified in the world coordinate system $\{W\}$. Therefore they need to be transformed to the robot frame $\{R\}$ :

$$
{ }^{W} z_{t, i}={ }^{W}\left[\begin{array}{l}
\alpha_{t, i}  \tag{5.63}\\
r_{t, i}
\end{array}\right] \rightarrow{ }^{R} z_{t, i}={ }^{R}\left[\begin{array}{l}
\alpha_{t, i} \\
r_{t, i}
\end{array}\right]
$$

According to figure (5.31), the transformation is given by

$$
\begin{align*}
\hat{z}_{i}(k+1) & =\left[\begin{array}{l}
R \\
\alpha_{t, i} \\
r_{t, i}
\end{array}\right]=h_{i}\left(z_{t, i}, \hat{p}(k+1 \mid k)\right) \\
& =\left[\begin{array}{c}
{ }^{W} \alpha_{t, i}-{ }^{W} \hat{\theta}(k+1 \mid k) \\
{ }^{W} r_{t, i}-\left({ }^{W} \hat{x}(k+1 \mid k) \cos \left({ }^{W} \alpha_{t, i}\right)+{ }^{W} \hat{y}(k+1 \mid k) \sin \left({ }^{W} \alpha_{t, i}\right)\right)
\end{array}\right] \tag{5.64}
\end{align*}
$$

and its Jacobian $\nabla h_{i}$ by


Figure 5.31
Representation of the target position in the world coordinate frame $\{W\}$ and robot coordinate frame $\{R\}$.

$$
\nabla h_{i}=\left[\begin{array}{lll}
\frac{\partial \alpha_{t, i}}{\partial \hat{x}} & \frac{\partial \alpha_{t, i}}{\partial \hat{y}} & \frac{\partial \alpha_{t, i}}{\partial \hat{\theta}}  \tag{5.65}\\
\frac{\partial r_{t, i}}{\partial \hat{x}} & \frac{\partial r_{t, i}}{\partial \hat{y}} & \frac{\partial r_{t, i}}{\partial \hat{\theta}}
\end{array}\right]=\left[\begin{array}{ccc}
0 & 0 & -1 \\
-\cos ^{W} \alpha_{t, i}-\sin ^{W} \alpha_{t, i} & 0
\end{array}\right]
$$

The measurement prediction results in predicted lines represented in the robot coordinate frame (figure 5.32). They are uncertain, because the prediction of robot position is uncertain.
4. Matching. For matching, we must find correspondence (or a pairing) between predicted and observed features (figure 5.33). In our case we take the Mahalanobis distance


Figure 5.32
Measurement predictions: Based on the map and the estimated robot position the targets (visible lines) are predicted. They are represented in the model space similar to the observations.


Figure 5.33
Matching: The observations (thick) and measurement prediction (thin) are matched and the innovation and its uncertainties are calculated.

$$
\begin{equation*}
v_{i j}^{T}(k+1) \cdot \Sigma_{I N, i j}^{-1}(k+1) \cdot v_{i j}(k+1) \leq g^{2} \tag{5.66}
\end{equation*}
$$

with

$$
\begin{align*}
v_{i j}(k+1) & =\left[z_{j}(k+1)-h_{i}\left(z_{t}, \hat{p}(k+1 \mid k)\right)\right] \\
& =\left[\begin{array}{c}
\alpha_{j} \\
r_{j}
\end{array}\right]-\left[\begin{array}{c}
{ }^{W} \alpha_{t, i}-{ }^{W} \hat{\theta}(k+1 \mid k) \\
{ }^{W} r_{t, i}-\left({ }^{W} \hat{x}(k+1 \mid k) \cos \left({ }^{W} \alpha_{t, i}\right)+{ }^{W} \hat{y}(k+1 \mid k) \sin \left({ }^{W} \alpha_{t, i}\right)\right)
\end{array}\right] \tag{5.67}
\end{align*}
$$

$$
\begin{equation*}
\Sigma_{I N, i j}(k+1)=\nabla h_{i} \cdot \Sigma_{p}(k+1 \mid k) \cdot \nabla h_{i}^{T}+\Sigma_{R, i}(k+1) \tag{5.68}
\end{equation*}
$$



Figure 5.34
Kalman filter estimation of the new robot position: By fusing the prediction of robot position (thin) with the innovation gained by the measurements (thick) we get the updated estimate $\hat{p}(k \mid k)$ of the robot position (very thick).
to enable finding the best matches while eliminating all other remaining observed and predicted unmatched features.
5. Estimation. Applying the Kalman filter results in a final pose estimate corresponding to the weighted sum of (figure 5.34)

- the pose estimates of each matched pairing of observed and predicted features;
- the robot position estimation based on odometry and observation positions.


### 5.7 Other Examples of Localization Systems

Markov localization and Kalman filter localization have been two extremely popular strategies for research mobile robot systems navigating indoor environments. They have strong formal bases and therefore well-defined behavior. But there are a large number of other localization techniques that have been used with varying degrees of success on commercial and research mobile robot platforms. We will not explore the space of all localization systems in detail. Refer to surveys such as [5] for such information.

There are, however, several categories of localization techniques that deserve mention. Not surprisingly, many implementations of these techniques in commercial robotics

