- Unequal floor contact (slipping, nonplanar surface, etc.).

Some of the errors might be *deterministic* (systematic); thus, they can be eliminated by proper calibration of the system. However, there are still a number of *nondeterministic* (random) errors that remain, leading to uncertainties in position estimation over time. From a geometric point of view, one can classify the errors into three types:

1. Range error: integrated path length (distance) of the robot's movement
   → sum of the wheel movements
2. Turn error: similar to range error, but for turns
   → difference of the wheel motions
3. Drift error: difference in the error of the wheels leads to an error in the robot's angular orientation

Over long periods of time, turn and drift errors far outweigh range errors, since their contribution to the overall position error is nonlinear. Consider a robot whose position is initially perfectly wellknown, moving forward in a straight line along the $x$-axis. The error in the $y$-position introduced by a move of $d$ meters will have a component of $d\sin\Delta\theta$, which can be quite large as the angular error $\Delta\theta$ grows. Over time, as a mobile robot moves about the environment, the rotational error between its internal reference frame and its original reference frame grows quickly. As the robot moves away from the origin of these reference frames, the resulting linear error in position grows quite large. It is instructive to establish an error model for odometric accuracy and see how the errors propagate over time.

### 5.2.4 An error model for odometric position estimation

Generally the pose (position) of a robot is represented by the vector

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} . \tag{5.1}$$

For a differential-drive robot (figure 5.3) the position can be estimated starting from a known position by integrating the movement (summing the incremental travel distances). For a discrete system with a fixed sampling interval $\Delta t$, the incremental travel distances $(\Delta x;\Delta y;\Delta\theta)$ are

$$\Delta x = \Delta s\cos(\theta + \Delta\theta/2) , \tag{5.2}$$

$$\Delta y = \Delta s\sin(\theta + \Delta\theta/2) , \tag{5.3}$$

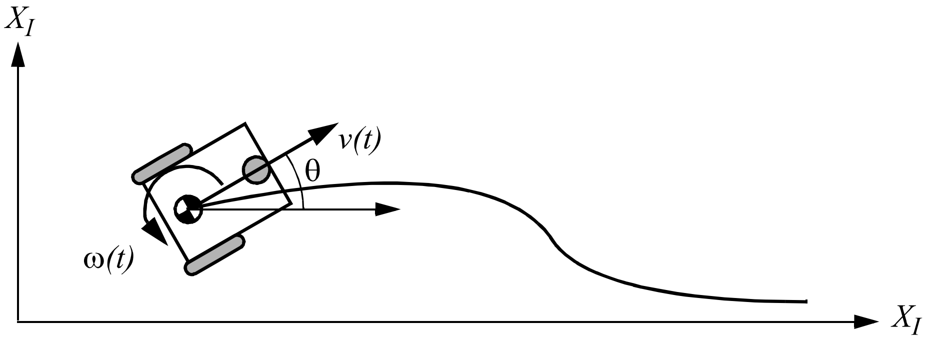

**Figure 5.3**
Movement of a differential-drive robot.

$$\Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} , \tag{5.4}$$

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} , \tag{5.5}$$

where

$(\Delta x;\Delta y;\Delta\theta)$ = path traveled in the last sampling interval;

$\Delta s_r;\Delta s_l$ = traveled distances for the right and left wheel respectively;

$b$ = distance between the two wheels of differential-drive robot.

Thus we get the updated position $p'$ :

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = p + \begin{bmatrix} \Delta s\cos(\theta + \Delta\theta/2) \\ \Delta s\sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s\cos(\theta + \Delta\theta/2) \\ \Delta s\sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} . \tag{5.6}$$

By using the relation for $(\Delta s;\Delta\theta)$ of equations (5.4) and (5.5) we further obtain the basic equation for odometric position update (for differential drive robots):

$$p' = f(x, y, \theta, \Delta s_r, \Delta s_l) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} . \tag{5.7}$$

As we discussed earlier, odometric position updates can give only a very rough estimate of the actual position. Owing to integration errors of the uncertainties of $p$ and the motion errors during the incremental motion $(\Delta s_r;\Delta s_l)$, the position error based on odometry integration grows with time.

In the next step we will establish an error model for the integrated position $p'$ to obtain the covariance matrix $\Sigma_{p'}$ of the odometric position estimate. To do so, we assume that at the starting point the initial covariance matrix $\Sigma_p$ is known. For the motion increment $(\Delta s_r;\Delta s_l)$ we assume the following covariance matrix $\Sigma_\Delta$:

$$\Sigma_\Delta = covar(\Delta s_r, \Delta s_l) = \begin{bmatrix} k_r|\Delta s_r| & 0 \\ 0 & k_l|\Delta s_l| \end{bmatrix} , \tag{5.8}$$

where $\Delta s_r$ and $\Delta s_l$ are the distances traveled by each wheel, and $k_r$, $k_l$ are error constants representing the nondeterministic parameters of the motor drive and the wheel-floor interaction. As you can see, in equation (5.8) we made the following assumptions:

- The two errors of the individually driven wheels are independent,[22]
- The variance of the errors (left and right wheels) are proportional to the absolute value of the traveled distances $(\Delta s_r;\Delta s_l)$.

These assumptions, while not perfect, are suitable and will thus be used for the further development of the error model. The *motion errors* are due to imprecise movement because of deformation of wheel, slippage, unequal floor, errors in encoders, and so on. The values for the error constants $k_r$ and $k_l$ depend on the robot and the environment and should be experimentally established by performing and analyzing representative movements.

If we assume that $p$ and $\Delta_{rl} = [\Delta s_r, \Delta s_l]^T$ are uncorrelated and the derivation of $f$ (equation [5.7]) is reasonably approximated by the first-order Taylor expansion (linearization), we conclude, using the error propagation law (see section 4.1.3.2),

[22] If there is more knowledge regarding the actual robot kinematics, the correlation terms of the covariance matrix could also be used.

$$\Sigma_{p'} = \nabla_p f \cdot \Sigma_p \cdot \nabla_p f^T + \nabla_{\Delta_{rl}} f \cdot \Sigma_\Delta \cdot \nabla_{\Delta_{rl}} f^T . \tag{5.9}$$

The covariance matrix $\Sigma_p$ is, of course, always given by the $\Sigma_{p'}$ of the previous step, and can thus be calculated after specifying an initial value (e.g., 0).

Using equation (5.7) we can develop the two *Jacobians*, $F_p = \nabla_p f$ and $F_{\Delta_{rl}} = \nabla_{\Delta_{rl}} f$ :

$$F_p = \nabla_p f = \nabla_p (f^T) = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s \sin(\theta + \Delta\theta/2) \\ 0 & 1 & \Delta s \cos(\theta + \Delta\theta/2) \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.10}$$

$$F_{\Delta_{rl}} = \begin{bmatrix} \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) & , \frac{1}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\sin\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) & , \frac{1}{2}\sin\left(\theta + \frac{\Delta\theta}{2}\right) - \frac{\Delta s}{2b}\cos\left(\theta + \frac{\Delta\theta}{2}\right) \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \tag{5.11}$$

The details for arriving at equation (5.11) are

$$F_{\Delta_{rl}} = \nabla_{\Delta_{rl}} f = \begin{bmatrix} \frac{\partial f}{\partial \Delta s_r} & \frac{\partial f}{\partial \Delta s_l} \end{bmatrix} = \ldots \tag{5.12}$$

$$\begin{bmatrix} \frac{\partial \Delta s}{\partial \Delta s_r}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2} - \sin\left(\theta + \frac{\Delta\theta}{2}\right)\frac{\partial \Delta\theta}{\partial \Delta s_r} & , \frac{\partial \Delta s}{\partial \Delta s_l}\cos\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2} - \sin\left(\theta + \frac{\Delta\theta}{2}\right)\frac{\partial \Delta\theta}{\partial \Delta s_l} \\ \frac{\partial \Delta s}{\partial \Delta s_r}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right)\frac{\partial \Delta\theta}{\partial \Delta s_r} & , \frac{\partial \Delta s}{\partial \Delta s_l}\sin\left(\theta + \frac{\Delta\theta}{2}\right) + \frac{\Delta s}{2}\cos\left(\theta + \frac{\Delta\theta}{2}\right)\frac{\partial \Delta\theta}{\partial \Delta s_l} \\ \frac{\partial \Delta\theta}{\partial \Delta s_r} & \frac{\partial \Delta\theta}{\partial \Delta s_l} \end{bmatrix} \tag{5.13}$$

and with

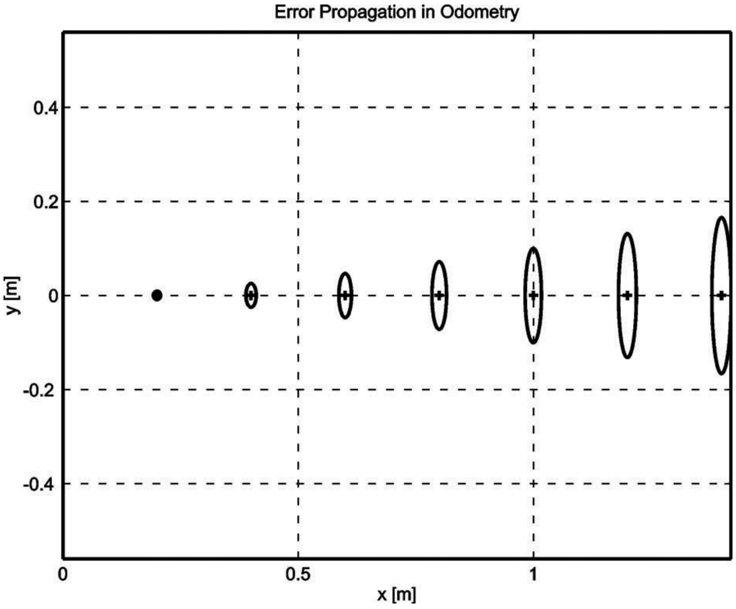

**Figure 5.4**
Growth of the pose uncertainty for straight-line movement: Note that the uncertainty in *y* grows much faster than in the direction of movement. This results from the integration of the uncertainty about the robot's orientation. The ellipses drawn around the robot positions represent the uncertainties in the *x,y* direction (e.g. $3\sigma$ ). The uncertainty of the orientation $\theta$ is not represented in the picture, although its effect can be indirectly observed.

$$\Delta s = \frac{\Delta s_r + \Delta s_l}{2} \quad ; \quad \Delta\theta = \frac{\Delta s_r - \Delta s_l}{b} \tag{5.14}$$

$$\frac{\partial \Delta s}{\partial \Delta s_r} = \frac{1}{2} \quad ; \quad \frac{\partial \Delta s}{\partial \Delta s_l} = \frac{1}{2} \quad ; \quad \frac{\partial \Delta\theta}{\partial \Delta s_r} = \frac{1}{b} \quad ; \quad \frac{\partial \Delta\theta}{\partial \Delta s_l} = -\frac{1}{b}, \tag{5.15}$$

we obtain equation (5.11).

Figures 5.4 and 5.5 show typical examples of how the position errors grow with time. The results have been computed using the error model presented earlier.

Once the error model has been established, the error parameters must be specified. One can compensate for deterministic errors properly calibrating the robot. However the error parameters specifying the nondeterministic errors can only be quantified by statistical (repetitive) measurements. A detailed discussion of odometric errors and a method for calibration and quantification of deterministic and nondeterministic errors can be found in [6]. A method for on-the-fly odometry error estimation is presented in [205].

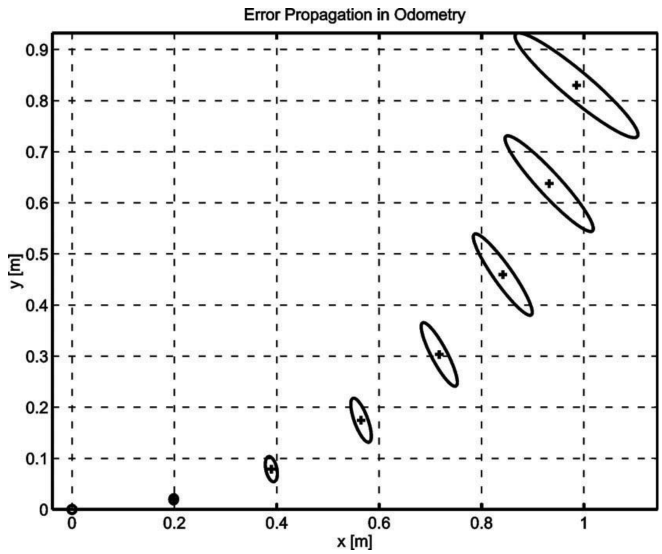

**Figure 5.5**
Growth of the pose uncertainty for circular movement ($r$ = const): Again, the uncertainty perpendicular to the movement grows much faster than that in the direction of movement. Note that the main axis of the uncertainty ellipse does not remain perpendicular to the direction of movement.

## 5.3 To Localize or Not to Localize: Localization-Based Navigation Versus Programmed Solutions

Figure 5.6 depicts a standard indoor environment that a mobile robot navigates. Suppose that the mobile robot in question must deliver messages between two specific rooms in this environment: rooms $A$ and $B$. In creating a navigation system, it is clear that the mobile robot will need sensors and a motion control system. Sensors are absolutely required to avoid hitting moving obstacles such as humans, and some motion control system is required so that the robot can deliberately move.

It is less evident, however, whether or not this mobile robot will require a *localization system*. Localization may seem mandatory in order to navigate successfully between the two rooms. It is through localizing on a map, after all, that the robot can hope to recover its position and detect when it has arrived at the goal location. It is true that, at the least, the robot must have a way of detecting the goal location. However, explicit localization with reference to a map is not the only strategy that qualifies as a goal detector.

An alternative, espoused by the behavior-based community, suggests that, since sensors and effectors are noisy and information-limited, one should avoid creating a geometric map for localization. Instead, this community suggests designing sets of behaviors that together result in the desired robot motion. Fundamentally, this approach avoids explicit reasoning about localization and position, and thus generally avoids explicit path planning as well.