International Journal of Artificial Intelligence and Machine Learning
Volume 10 • Issue 1 • January-June 2020

Sensor Fusion of Odometer, Compass and Beacon Distance for Mobile Robots

Rufus Fraanje, The Hague University of Applied Sciences, Delft, Netherlands
René Beltman, Lely, Maassluis, Netherlands
Fidelis Theinert, The Hague University of Applied Sciences, Delft, Netherlands
Michiel van Osch, Fontys University of Applied Sciences, Eindhoven, Netherlands
Teade Punter, Fontys University of Applied Sciences, Eindhoven, Netherlands
John Bolte, The Hague University of Applied Sciences, Delft, Netherlands

ABSTRACT

The estimation of the pose of a differential drive mobile robot from noisy odometer, compass, and beacon distance measurements is studied. The estimation problem, which is a state estimation problem with unknown input, is reformulated into a state estimation problem with known input and a process noise term. A heuristic sensor fusion algorithm solving this state-estimation problem is proposed and compared with the extended Kalman filter solution and the Particle Filter solution in a simulation experiment.

KEYWORDS

Differential Drive Robot, Extended Kalman Filter, Navigation, Odometry, Particle Filter, Sensor Fusion

INTRODUCTION

In navigation of mobile robots information from multiple sensors need to be used to reliably determine the position and orientation, i.e. the pose, of the mobile robot (Thrun & Burgard, 2005; Dudek & Jenkin, 2010; Gustafsson, 2010). In outdoor robotics applications usually GPS or DGPS measurements are used to compensate for the drift in the position estimation based on odometry, see e.g. (Ohno et al., 2004). For autonomous driving on high-ways and in urban environments other sensors, such as Lidar, Radar and/or camera's, are being used as well, see e.g. (Thrun & Burgard, 2005). For navigation in indoor applications or near tall buildings, where GPS or other GNSS has weak or no coverage, other beacons can be used e.g., based on ultrasound transmission time (Wijk & Christensen, 2000) or received signal strength (RSS) of visible light (Plets et al., 2017) and transmission time of ultra-wideband (UWB) transmission (Dabove et al., 2018). For indoor navigation also beacon-less methods are being used, that rely on simultaneous localisation and mapping (SLAM) (Thrun & Burgard, 2005) and often make use of optical sensors observing the robots environment, such as Lidar and/or (RGBd or stereo) camera's.

DOI: 10.4018/IJAIML.2020010101

In dirty and/or dusty working environments optical sensors may not be effective and ultrasonic or UWB beacon-based techniques are needed to compensate for drift in (semi) indoor mobile robot navigation. There are several approaches that can be taken:

- **Odometry based:** The pose of the robot is determined only by odometry, and optionally including compass and/or IMU sensors;
- **Beacon based:** Odometry information is not used for navigation, only triangulation based on two or more time of arrival (TOA) or three or more-time difference of arrival (TDOA) measurements;
- **Beacon based resetting of odometry:** The position of the robot determined by odometry is reset to the position determined by a beacon based (TOA or TDOA) method (the resetting is usually at a lower rate than the odometry update rate);
- **Sensor fusion of beacon and odometry based measurements:** The measurements from the odometry, optional compass and IMU sensors and the beacon-based sensors are fused according to some sensor fusion algorithm to provide an estimate of the robots pose.

In this paper, the latter approach of fusing the beacon and odometry sensor measurements, including a compass sensor, is used to achieve an estimate of the robots pose. Because the intended application is for mobile robots in dirty and/or dusty environments the choice has been made to focus on sensor fusion of odometry, compass and UWB beacon distance measurements. Various sensor fusion algorithms are evaluated, a heuristic approach, the extended Kalman filter and the particle filter, see e.g. (Thrun & Burgard, 2005; Dudek & Jenkin, 2010; Gustafsson, 2010). The algorithms are compared in a simulation experiment. Parts of this paper, especially the algorithm presentation, have been published in (Fraanje et al., 2019) as a conference publication. In addition, the current paper discusses various implementation issues, gives directions for extensions, such as the multi-beacon case, and the section on the simulation experiments and their discussion is extended and fully revised.

PROBLEM ANALYSIS

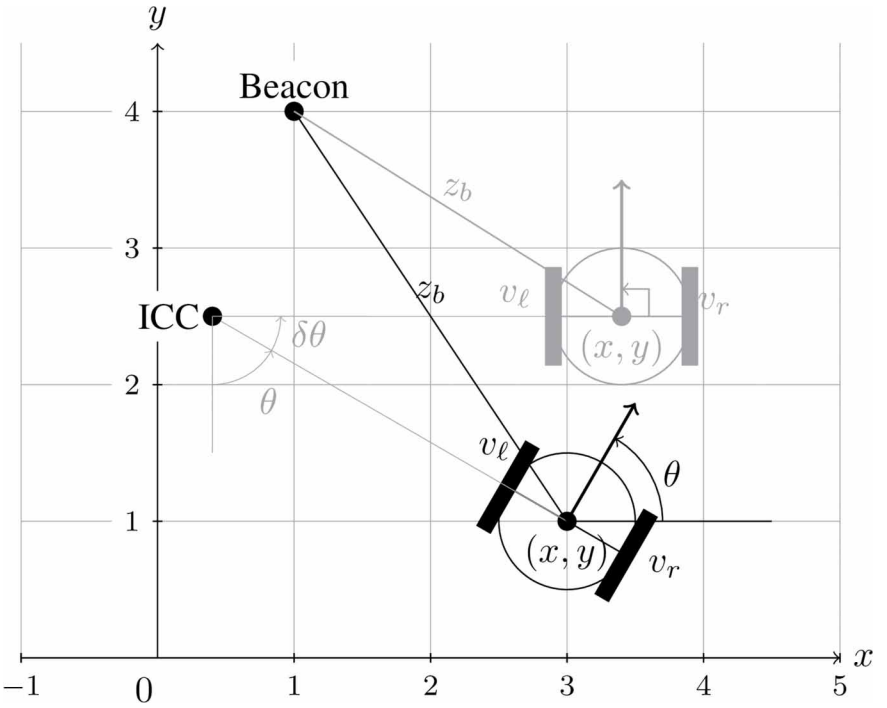
Figure 1 shows the schematic of a differential drive mobile robot at position $\left(X_k, Y_k\right)$ and heading θ_k relative to some coordinate frame, where $k = 0,1,...$ refers to a discrete time index, i.e. $t = k\delta t$ where δt the time difference between two time steps. The left and right wheel velocities, denoted by $v_{l,k}$ and $v_{r,k}$ respectively, are given by:

$$v_{l/r,k} = 2\pi r_{l/r} n_{l/r,k} \tag{1}$$

where r_l and r_r wheel radii and $n_{l,k}$ and $n_{r,k}$ the number of rotations per second at time index k of the left and right wheel respectively. Note that usually the left and right wheel radii are equal but this is not necessary. The odometry equations that yield the update to the mobile robot pose, i.e. position and heading, have been derived in many textbooks, see e.g. (Dudek & Jenkin, 2010). Here, we present an alternative form, that can be derived using some additional trigonometry:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} \delta x_k \\ \delta y_k \\ \delta \theta_k \end{bmatrix} \tag{2}$$

Figure 1. Schematic of the differential drive mobile robot with the distance to a beacon being measured. ICC is the Instantaneous Center of Curvature. Dependency on the time index k is suppressed for reasons of clarity.



where:

$$\begin{bmatrix}\delta x_k\\ \delta y_k\end{bmatrix} = s_k \begin{bmatrix}\operatorname{sinc}\left(\delta\theta_k\right) & -\sin\left(\delta\theta_k / 2\right)\\ \sin\left(\delta\theta_k / 2\right) & \operatorname{sinc}\left(\delta\theta_k\right)\end{bmatrix}\begin{bmatrix}\cos\left(\theta_k\right)\\ \sin\left(\theta_k\right)\end{bmatrix},$$
$$\delta\theta_k = \delta t\left(V_{r,k} - V_{\ell,k}\right)/b,$$
$$s_k = \delta t\left(V_{r,k} + V_{\ell,k}\right)/2 \tag{3}$$

and b the baseline, i.e. the distance between the wheels and $\operatorname{sinc}\left(\delta\theta_k\right) = \sin\left(\delta\theta_k\right)/\delta\theta_k$.

Note that this expression holds for all $v_{r,k}$ and $v_{l,k}$ and it is not necessary to distinguish the cases $v_{r,k}=v_{l,k}$ and $v_{r,k} \neq v_{l,k}$ from each other, which simplifies the implementation of the odometry model as well as the calculation of linearized models in the extended Kalman filter. Because often in practice the angle update between two-time instances are small, i.e. $\left|\delta\theta_k\right| \ll 1, \operatorname{sinc}\left(\delta\theta_k\right)$ can be approximated very well by the first few terms of its power series expansion about 0:

$$\operatorname{sinc}\left(\delta\theta_k\right) = 1 - \frac{\left(\delta\theta_k\right)^2}{3!} + \frac{\left(\delta\theta_k\right)^4}{5!} - \ldots \tag{4}$$

which provides a computationally efficient implementation.

The measurement of the distance to a beacon, which position $\left(x_b, y_b\right)$ is known, is given by:

$$z_{b,k} = \sqrt{(x_b - x_k)^2 + (y_b - y_k)^2 + v_{b,k}} \tag{5}$$

where $v_{b,k}$ represents noise that deteriorates the measurement of the distance, and can be caused by scattering, fading, propagation time errors, etc. Note, that this paper focuses on the case of a single beacon, but the equation can be vectorized straightforwardly for multiple beacons. Also the position of the beacons need not to be fixed, and even need not to be in the same plane as the mobile robot, as long as the beacon positions are known at each time instance and Equation (5) is adjusted accordingly.

The measurements of the left and right wheel speeds, determined by wheel encoders or tachometers, and the heading, determined by a compass or a gyroscope, are stored in the following observation vector:

$$z_{o,k} = \begin{bmatrix} v_{r,k} & v_{l,k} & \theta_k \end{bmatrix}^T + \begin{bmatrix} v_{vr,k} & v_{vl,k} & v_{\theta,k} \end{bmatrix}^T \tag{6}$$

where $v_{vr,k}, v_{vl,k}$ and $v_{\theta,k}$ represent measurement noises.

For notational convenience, the following definitions are being made. The pose is denoted by:

$$q_k = \begin{bmatrix} x_k & y_k & \theta_k \end{bmatrix}^T \tag{7}$$

and the command vector is denoted by:

$$u_k = \begin{bmatrix} \delta\theta_k & s_k \end{bmatrix}^T \tag{8}$$

which is directly determined by the left and right wheel velocities according to:

$$\begin{bmatrix} \delta\theta_k \\ s_k \end{bmatrix} = \delta t \begin{bmatrix} 1/b & -1/b \\ 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} V_{r,k} \\ V_{l,k} \end{bmatrix} \tag{9}$$

Then, the pose update equation is written by:

$$q_{k+1} = f\left(q_k, u_k\right) \tag{10}$$

where the function f can be easily derived from (2), (3):

$$f\left(q,u\right) = q + \begin{bmatrix} u(2) \begin{bmatrix} \operatorname{sinc}\left(u(1)\right) & -\operatorname{sinc}\left(u(1)/2\right) \\ \operatorname{sinc}\left(u(1)/2\right) & \operatorname{sinc}\left(u(1)\right) \end{bmatrix} \begin{bmatrix} \cos\left(q(3)\right) \\ \sin\left(q(3)\right) \end{bmatrix} \\ u(1) \end{bmatrix} \tag{11}$$

where *u(i)* and *q(i)* refer to the i^{th} element of the vectors *u* and *q* respectively; this notation will be adopted for indexing elements in other vectors in this paper as well. For reasons of clarity the dependency on the time index *k* is suppressed.

All measurements, the beacon distance (5) and the wheel velocities and heading (6), at time instance *k* are stored in one measurement vector:

$$z_k = \begin{bmatrix} z_{b,k} & z_{o,k}^T \end{bmatrix}^T \tag{12}$$

When multiple beacons are being used, $z_{b,k}$ will be a vector instead of a scalar and to achieve a proper vector stacking $z_{b,k}$ should be replaced by $z_{b,k}^T$ in Equation (12). Then, the measurement equation can be compactly written as:

$$z_k = g\left(q_k, u_k\right) + \nu_k \tag{13}$$

where V_k is the vector stacking of all the measurement noises, and the function *g* is defined as:

$$g\left(q,u\right) = \begin{bmatrix} \sqrt{x_b - q\left(1\right))^2 + (y_b - q\left(2\right))^2} \\ \frac{1}{\delta t}\begin{bmatrix} b/2 & 1 \\ -b/2 & 1 \end{bmatrix}\begin{bmatrix} u\left(1\right) \\ u\left(2\right) \end{bmatrix} \\ q\left(3\right) \end{bmatrix} \tag{14}$$

Then, the problem is to (recursively) estimate the pose q_k given the noisy measurements z_k subject to Equations (10) and (13). Note, that this is an estimation problem with an unknown input u_k, because the measured wheel velocities, z_k (2) and $z_k\left(3\right)$, are distorted by noise from which only a noise deteriorated estimate of u_k can be determined. One possibility would be to augment the state vector with the unknown input u_k and adopting a hypermodel for the time-update of u_k. Then, the augmented state $\begin{bmatrix} q_k^T & u_k^T \end{bmatrix}^T$ can be estimated from the measurements z_k, subject to adjusted versions of Equations (10) and (13). In this paper, a simpler approach is taken to deal with the *unknown* input u_k, which is estimated first from the measurements:

$$\hat{u}_k = \delta t \begin{bmatrix} 1/b & -1/b \\ 1/2 & 1/2 \end{bmatrix}\begin{bmatrix} z_k\left(2\right) \\ z_k\left(3\right) \end{bmatrix} \tag{15}$$

Then the pose update equation can be written in terms of the estimate $\hat{u}_k$ rather than the true u_k where a process noise term η_k accounts for the error in the estimate:

$$q_{k+1} = f\left(q_k, \hat{u}_k\right) + \eta_k \tag{16}$$

where $\eta_k = f\left(q_k, u_k\right) - f\left(q_k, \hat{u}_k\right)$. For small values of $u_k\left(1\right)$ and $\hat{u}_k\left(1\right)$, i.e. $\delta\theta_k$ and $\hat{\delta}\theta_k$ are small, it can be shown that:

$$\eta_k = f\left(q_k, u_k\right) - f\left(q_k, \hat{u}_k\right) \approx \begin{bmatrix} \frac{\delta t}{2}\left(\nu_{r,k} + \nu_{l,k}\right)\begin{bmatrix}\cos\left(q_k\left(3\right)\right)\\ \sin\left(q_k\left(3\right)\right)\end{bmatrix} \\ \frac{\delta t}{b}\left(\nu_{r,k} - \nu_{l,k}\right) \end{bmatrix} \tag{17}$$

This approximation of η_k will be used later to estimate its covariance for use in the extended Kalman filter. Because the wheel speed measurements $z_k\left(2\right)$ and $z_k\left(3\right)$ have been used already in determining $\hat{u}_k$ they are not considered as outputs anymore, and we introduce the reduced output vector consisting of the beacon distance (or the vector with beacon distances for multiple beacons) and the heading:

$$z_k^{'} = \begin{bmatrix} z_k\left(1\right) & z_k\left(4\right)\end{bmatrix}^T = g'\left(q_k\right) + \nu_k^{'} \tag{18}$$

where:

$$g'\left(q\right) = \begin{bmatrix} \sqrt{\left(x_b - q\left(1\right)\right)^2 + \left(y_b - q\left(2\right)\right)^2} \\ q\left(3\right)\end{bmatrix} \tag{19}$$

where the vector with the true beacon distance and the heading, and $v_k^{'} = \begin{bmatrix} v_k\left(1\right) & v_k\left(4\right)\end{bmatrix}^T$ the vector with the beacon distance noise and the heading noise respectively. With these adjustments, there is a process noise term η_k introduced in the pose update equation, but the unknown input in the estimation problem is removed.

SENSOR FUSION ALGORITHMS

Heuristic Approach

This approach is rather straightforward, and its implementation is efficient. The idea is to first estimate the pose by updating the odometry equations and fuse the heading and then to fuse this estimate with the measurement from the distance to the beacon. For the case of multiple beacons, the latter step can be repeated for each beacon distance measurement. The procedure for a single beacon is outlined as follows:

1. Set the initial pose estimate $\hat{q}_{0|0}$ and the initial command vector estimate $\hat{u}_0$;
2. Iterate for $k = 1, 2,...$:
   Perform a time update of the pose estimate:

$$\hat{q}_{k|k-1} = f\left(\hat{q}_{k-1|k-1}, \hat{u}_{k-1}\right) \tag{20}$$

Determine the predicted output:

$$\hat{z}_k^{'} = g'\left(\hat{q}_{k|k-1}\right) \tag{21}$$

3. Measure z_k and determine $z_k^{'}$ and $\hat{u}_k$;
4. Make a weighted average of the heading (fusion step):

$$\hat{q}_{k|k}\left(3\right) = \frac{C_{\theta,p}^{-1}\hat{z}_k^{'}\left(2\right) + C_{\theta,m}^{-1}z_k^{'}\left(2\right)}{C_{\theta,p}^{-1} + C_{\theta,m}^{-1}} \tag{22}$$

where $C_{\theta,p}$ the variance of the error in the heading estimate $z_k^{'}\left(2\right)$ and $C_{\theta,m}$ the variance of the noise in measurement $z_k^{'}\left(2\right)$.

5. Update the position by fusing the beacon distance measurement:

    a. Calculate the angular position of the mobile robot relative to the beacon:

$$\widehat{\Phi}_{k|k-1} = atan2\left(\hat{q}_{k|k-1}\left(2\right) - y_b\hat{q}_{k|k-1}\left(1\right) - x_b\right) \tag{23}$$

    b. Make a weighted average of the distance to the beacon (fusion step):

$$\hat{z}_{k|k}^{'}\left(1\right) = \frac{C_{b,p}^{-1}\hat{z}_k^{'}\left(1\right) + C_{b,m}^{-1}z_k^{'}\left(1\right)}{C_{b,p}^{-1} + C_{b,m}^{-1}} \tag{24}$$

where $C_{b,p}$ the variance of the error in the estimation $z_k^{'}\left(1\right)$ of the distance to the beacon and $C_{b,m}$ the variance of the noise measurement $z_k^{'}\left(1\right)$.

    c. Calculate the position estimate:

$$\begin{bmatrix}\hat{q}_{k|k}\left(1\right)\\ \hat{q}_{k|k}\left(2\right)\end{bmatrix} = \begin{bmatrix}x_b\\ y_b\end{bmatrix} + \hat{z}_{k|k}\left(1\right)\begin{bmatrix}\cos\left(\widehat{\Phi}_{k|k-1}\right)\\ \sin\left(\widehat{\Phi}_{k|k-1}\right)\end{bmatrix} \tag{25}$$

The algorithm heavily relies on the fusion equation of two estimates, each with its own error variance, see e.g. (Gustafsson, 2010). The implementation is rather straightforward, but the difficulty in the practical use is to determine the variances $C_{\theta,p}$, $C_{\theta,m}$ and $C_{b,p}$, $C_{b,m}$. In fact, only the ratio between $C_{\theta,p}$ and $C_{\theta,m}$ and between $C_{b,p}$ and $C_{b,m}$ needs to be determined, which involves determining the 'trust' in the model prediction versus the 'trust' in the measurement.

When there are multiple beacons, step 5 above, can be repeated for each beacon. Alternatively, the position of the mobile robot can be estimated by solving a triangulation problem using the beacon

distance measurements and then fusing this estimate with the predicted position estimation resulting from the time update Equation (20).

Extended Kalman Filter

The extended Kalman filter is obtained by applying the Kalman filter to the linearized pose update and measurement equation, i.e. the functions f and g'. The gradients of f and g' are given by (note, we suppress the dependency on the time index k):

$$A = \frac{\partial f}{\partial q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -u(2)\left(s_{12}c_{q3} + s_{c1}s_{q3}\right) & u(2)\left(s_{c1}c_{q3} - s_{12}s_{q3}\right) & 1 \end{bmatrix} \tag{26}$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} u(2)\left(\propto c_{q3} - c_{12}s_{q3} / 2\right) & s_{c1}c_{q3} - s_{12}s_{q3} \\ u(2)\left(c_{12}c_{q3} / 2 + \propto s_{q3}\right) & s_{12}c_{q3} + s_{c1}s_{q3} \\ 1 & 0 \end{bmatrix}$$

where:

$$\begin{aligned} s_{c1} &= \operatorname{sinc}\left(u(1)\right), \quad s_{12} = \sin\left(u(1) / 2\right) \\ c_{12} &= \cos\left(\frac{u(1)}{2}\right), \quad s_{q3} = \sin\left(q(3)\right) \\ c_{q3} &= \cos\left(q(3)\right), \quad \propto = \frac{u(1)\cos\left(u(1)\right) - \sin\left(u(1)\right)}{u^2(1)} \end{aligned} \tag{27}$$

and:

$$C = \frac{\partial g^{'}}{\partial g} = \begin{bmatrix} \dfrac{-\left(x_b - q(1)\right)}{z_b} & \dfrac{-\left(y_b - q(2)\right)}{z_b} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{28}$$

where:

$$z_b - \sqrt{\left(x_b - q(1)\right)^2 + \left(y_b - q(2)\right)^2} \tag{29}$$

Then, the nonlinear model of the differential drive mobile robot can be *approximated* by the following linear model:

$$\begin{aligned} q_{k+1} &= A_k q_k + B_k \hat{u}_k + n_k^{'} \\ z_k^{'} &= C_k^{'} q_k + V_k^{'} \end{aligned} \tag{30}$$

Assume that the process noise $n_k^{'}$ and the measurement noise $V_k^{'}$ can be approximated by zero-mean independent stochastic noise processes with correlation:

$$E\left(\begin{bmatrix} n_k^{'} \\ V_i^{'} \end{bmatrix}\begin{bmatrix} n_i^{'} \\ V_i^{'} \end{bmatrix}^T\right) = \begin{bmatrix} Q_k^i & 0 \\ 0 & R_k^{'} \end{bmatrix}\delta\left(k-i\right) \tag{31}$$

where $\delta\left(k\right)$ the Kronecker delta function, i.e. $\delta\left(k\right)=1$ and $\delta\left(k\right)\neq 0$ for $k\neq 0$. Under the assumption that $n_k^{'}=n_k$ and that $q_k(3)$ can be approximated by its estimate $\hat{q}_k\left(3\right)$ (i.e. neglect to contribution from the estimation error in the heading $q_k(3)$), we can approximate $Q_k^{'}$ and $R_k^{'}$ by:

$$Q_k^{'} = \left(\delta t\right)^2\left(Q_{v,r}+Q_{v,l}\right)\begin{bmatrix} \cos^2\left(\hat{q}_k\left(3\right)\right)/4 & \sin\left(\hat{q}_k\left(3\right)/2\right)/8 & \cos\left(\hat{q}_k\left(3\right)\right)/\left(2b\right) \\ \sin\left(\hat{q}_k\left(3\right)/2\right)/8 & \sin^2\left(\hat{q}_k\left(3\right)\right)/4 & \sin\left(\hat{q}_k\left(3\right)\right)/\left(2b\right) \\ \cos\left(\hat{q}_{\boldsymbol{k}}\left(3\right)\right)/\left(2b\right) & \sin\left(\hat{q}_k\left(3\right)\right)/\left(2b\right) & 1/b^2 \end{bmatrix} \tag{32}$$

and:

$$R_k^{'} = \begin{bmatrix} Q_b & 0 \\ 0 & Q_\theta \end{bmatrix} \tag{33}$$

where $Q_{v,r}$ and $Q_{v,l}$ are the variances of the measurement noises $v_{vr,k}$ and $v_{vl,k}$ on the right and left wheel velocities respectively, and are assumed to be zero-mean and independent; moreover Q_b and Q_θ are the variances of the noise $v_{b,k}$ on the beacon distance measurement and the noise $v_{\theta,k}$ on the heading measurement respectively, also assumed to be zero-mean and independent.

The estimation of the extended Kalman filter is given (see e.g., (Gustafsson, 2010)):

1. Set the initial pose estimate $\hat{q}_{0|0}$, the initial pose estimation error covariance matrix $P_{0|0}$ and the initial command vector estimate $\hat{u}_0$;
2. Iterate for k = 1,2,…:
   a. Calculate A_{k-1} and B_{k-1} $\hat{q}_{k-1|k-1}$ and $\hat{u}_{k-1}$;
   b. Perform the time update of the pose estimate:

$$\hat{q}_{k|k-1} = f\left(\hat{q}_{k-1|k-1}, \hat{u}_{k-1}\right) \tag{34}$$

   c. Calculate $Q_k^{'}$ using Equation (32) using $\hat{q}_{k|k-1}$ and the pose prediction error covariance:

$$P_{k|k-1} = A_{k-1} P_{k-1|k-1} A_{k-1}^T + Q_k^{'} \tag{35}$$

d. Calculate $C_k^{'}$ using $\hat{q}_{k|k-1}$;
e. Measure z_k and determine $z_k^{'}$ and $\hat{u}_k$;
f. Determine the predicted output:

$$\hat{z}_k^{'} = g'\left(\hat{q}_{k|k-1}\right) \tag{36}$$

and the innovation:

$$e_k^{'} = z_k^{'} - \hat{z}_k^{'} \tag{37}$$

and the Kalman gain:

$$K_k^{'} = P_{k|k-1} C_k^{'T} \left(C_k^{'} P_{k|k-1} C_k^{'T} + R_k^{'}\right)^{-1} \tag{38}$$

g. Perform a measurement update of the pose estimate:

$$\hat{q}_{k|k} = \hat{q}_{k|k-1} + K_k^{'} e_k^{'} \tag{39}$$

and the pose estimation error covariance:

$$P_{k|k} = P_{k|k-1} - K_k^{'} C_k^{'} P_{k|k-1} \tag{40}$$

Note, that the pose prediction and the output prediction are based on the non-linear model, i.e. the functions *f* and *g'*, and the linearized model is only used in calculating the pose prediction and estimation covariances and the Kalman gain.

Particle Filter

Whereas the extended Kalman filter assumes that the statistics of the pose can be approximated by a Gaussian probability density function, this is most likely not the case, especially because of the nonlinearity of the odometry equations. The particle filter, see e.g. (Gustafsson, 2010), relaxes this assumption, by sampling the probability density function by a number of so called particles representing an estimate of the pose, each with a weight, representing the probability of that particle. When a new measurement is available, each weight is updated by determining the probability of the predicted output of the particle given the actual measurement. Therefore, as in the case of the extended Kalman filter, we assume that the noise term $v_k^{'}$ is a zero-mean Gaussian distributed white noise signal with known covariance $R_k^{'}$. Then, the probability density function of the measurement noise $v_k^{'}$ is given by:

$$P_{v',k}\left(\xi\right)=\frac{e^{-\frac{1}{2}\xi^T R_{v',k}\xi}}{2\pi\sqrt{\left|R_{v',k}\right|}} \tag{41}$$

Then the Particle filter algorithm for estimating the pose is given by:

1. Choose (different) initial values for the N_p particles $\left\{\hat{q}^i_{0|0}\right\}_{i=1}^{N_p}$;
2. Iterate for $k = 1, 2, \ldots$:
   a. Perform a time update of the pose estimate particles:

$$\hat{q}^i_{k|k-1}=f\left(\hat{q}^i_{k-1|k-1},\hat{u}_{k-1}\right),\quad i=1,\ldots N_p \tag{42}$$

   b. Measure z_k and determine $z_k^{'}$ and $\hat{u}_k$;
   c. Determine for each particle the predicted output:

$$\hat{z}_k^{'i}=g'\left(\hat{g}^i_{k|k-1}\right),\quad i=1,\ldots N_p \tag{43}$$

   d. Calculate the weights with the probability density function of the noise on $z_k^{'}$:

$$w^i_{k|k}=\frac{p_{v'}\left(z_k^{'}-\hat{z}_k^{'i}\right)}{c_k},\quad i=1,\ldots,N_p \tag{44}$$

where $c_k=\sum_{i=1}^{N_p}p_{v'}\left(z_k^{'}-\hat{z}_k^{'i}\right)$ the normalization weight.

   e. Calculate the pose estimate by:

$$\hat{q}_{k|k}=\sum_{i=1}^{N_p}w^i_{k|k}\hat{q}^i_{k|k-1} \tag{45}$$

   f. Take samples $\left\{\hat{q}^i_{k|k}\right\}_{i=1}^{N_p}$ with replacement from the set $\left\{\hat{q}^i_{k|k-1}\right\}_{i=1}^{N_p}$ where the probability to take sample *i* is $w^i_{k|k}$;
   g. To prevent particle depletion (because of resampling with replacement more and more particles may become equal), a small amount of noise is added:

$$\hat{q}^i_{k|k-1}\leftarrow\hat{q}^i_{k|k-1}+\zeta_k \tag{46}$$

where ζ_k is a realization of a zero-mean (Gaussian) random vector with a small covariance, e.g. ϵI where ϵ a small positive number, say $\epsilon=10^{-4}$.

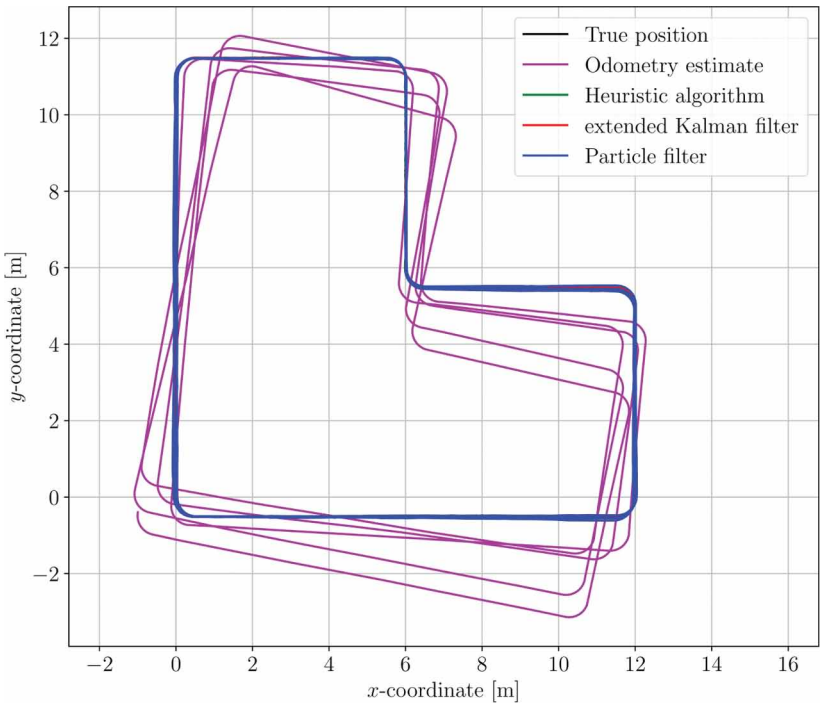
Simulation Experiment

The algorithms of the previous section are compared in a simulation experiment of a differential drive robot with baseline b=0.5 m, left and right wheel radii of $r_l = r_r = 0.5$ m and a time update rate of $\delta t = 0.02s$, i.e. a sample frequency of 50 Hz. The differential drive is commanded to drive five times along a closed rectangular shaped path of 46.7 m with a constant speed of about 1.6 m/s as illustrated in Figure 2. The starting point of the robot is at the origin, i.e. (0,0) m with heading 90° with respect to the x-axis. The beacon is at position (1, 7.5) m.

The measurement noises are assumed to be zero-mean, independent and Gaussian distributed with constant variances. The variance of the beacon distance measurement noise is $Q_{b,m} = 0.001$ m^2, of the right and left wheel velocity measurement noise is $Q_{v,r} = Q_{v,l} = 0.001$ m^2/s^2 and of the heading measurement noise is $Q_{\theta,m} = 0.001$ (°)^2. In addition to the measurement noise, we add a small constant bias on the left wheel velocity measurement of 10^{-3} m/s, to simulate the effect of a systematic error. This bias adds about 14cm tot the measured traveled distance of the left wheel, which is about 0.06% of the total traveled distance.

For the heuristic approach, the variances of the prediction errors are set to be 10 times smaller than those of the measurement noises, i.e. $Q_{\theta,p} = 0.0001$ (°)^2 and $Q_{b,p} = 0.0001$ m^2, to give more weight to the model based prediction, which turned out to give a smoother estimate. For the extended Kalman filter the initial pose estimate covariance matrix is set to $P_{0|0} = 10^{-5} I_3$, which is determined by a bit of trial and error. The covariances Q_k' and R_k' are determined as in the Extended Kalman Filter subsection above. For the Particle filter $N_p = 100$ particles have been used and the scaling

Figure 2. Example trajectory containing a rectangular path that is followed five times; the true position and the position estimated by the Heuristic algorithm and the extended Kalman filter are behind the estimation of the Particle filter (blue), the drifting in the odometry estimate (pink) is clearly visible



parameter of the spurious noise on the particles to prevent particle depletion was chosen as $\epsilon = 10^{-6}$. We have performed $N_{exp} = 100$ experiment, where each experiment consist of a generation of realizations of the measurement noises and sensor signals and a run of the various estimation algorithms, including the estimate solely based on odometry (i.e., only left and right wheel velocities and the pose update Equations (2) and (3)).

Figure 2 shows an example of the estimated trajectories in one experiment. We observe that the estimation algorithms, heuristic algorithm (green), extended Kalman filter (red) and Particle filter (blue) are perform relatively well, but the odometry estimate (pink) deviates more and more, as is expected since no feedback of measured quantities is being used. The drifting of the odometry is caused by the noise in the left and right wheel speed velocities. The small bias of 10^{-3} m/s on the left wheel speed measurement also contributes to the error in the odometry estimate but is not dominant here; removing this bias resulted in similar odometry estimation errors.

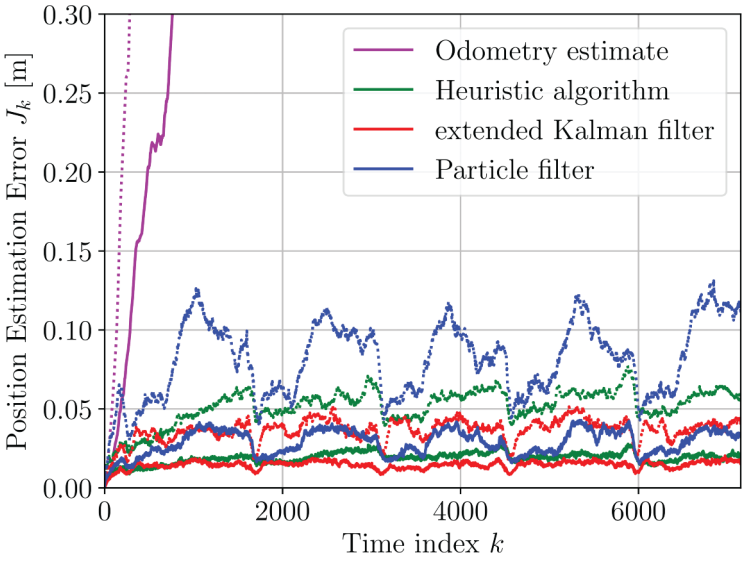
Figures 3-6 show the median, i.e. 50-percentile, (solid curves) and the 95-percentile (dotted curves) of the error in the position estimate, i.e.:

$$J_k = \sqrt{\left(q_k\left(1\right) - \hat{q}_k\left(1\right)\right)^2 + \left(q_k\left(2\right) - \hat{q}_k\left(2\right)\right)^2} \tag{47}$$

The scale on the y-axis has been chosen such that the difference on the well performing algorithms can be compared.

Figure 3 shows that the position estimate achieved by the extended Kalman filter performs the best, closely followed by the heuristic algorithm. The estimate obtained by the Particle filter is slightly worse, which may be explained by the fact that a limited number of particles $N_p = 100$ has been used. The results in Figure 3 are obtained by using feedback from the beacon distance and the heading measurements. To study the influence of the contribution of each sensor reading separately we also performed measurements running the estimation algorithms only using feedback from the beacon

Figure 3. Median (solid lines) and 95-percentile (dotted lines) of the error in position estimate obtained by odometry (pink), the heuristic algorithm (green), the extended Kalman filter (red) and the Particle filter (blue); the latter three use both beacon distance and heading measurements



distance (green curves in Figure 4-6) and only from the heading (red curves in Figures 4-6). The blue curves in Figures 4-6 show the estimation errors achieved when using both sensors (as in Figure 3).

We see that only using the beacon distance and no heading measurement resulted in large and diverging estimation errors. Only taking heading into account resulted in less bad, but still in a diverging performance for all estimation algorithms. This divergence is caused by the small bias on the left wheel velocity measurement. When we removed this bias, the estimation based on heading measurements only is stable and only slightly worse than

Figure 4. Median (solid lines) and 95-percentile (dotted lines) of the error in position estimate obtained by the heuristic algorithm using both beacon distance and heading (blue), only beacon distance (green) and only heading (red)

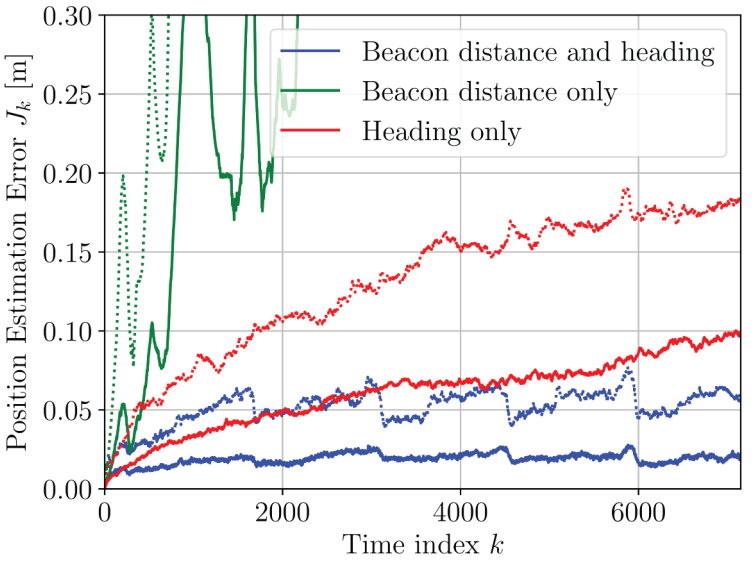


Figure 5. Median (solid lines) and 95-percentile (dotted lines) of the error in position estimate obtained by the extended Kalman filter using both beacon distance and heading (blue), only beacon distance (green) and only heading (red)

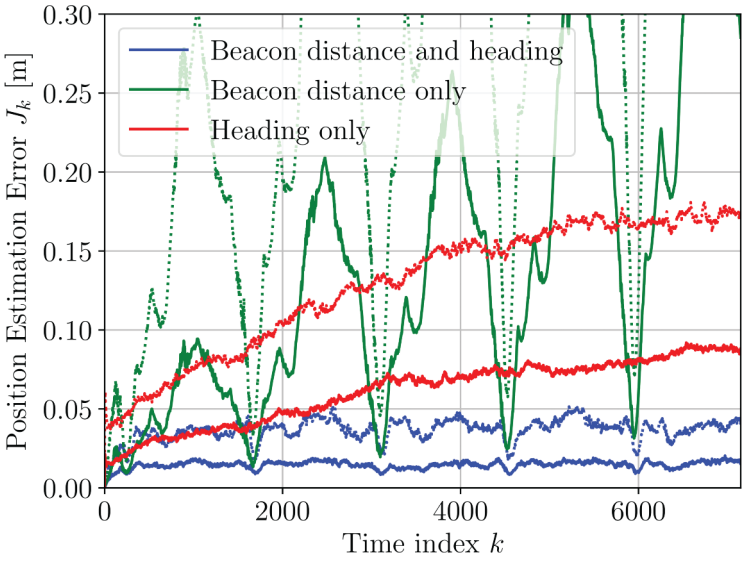
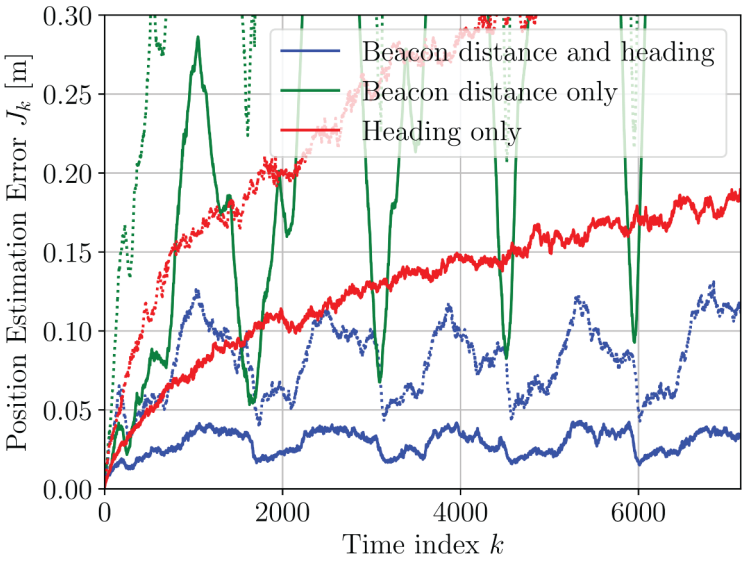


Figure 6. Median (solid lines) and 95-percentile (dotted lines) of the error in position estimate obtained by the Particle filter using both beacon distance and heading (blue), only beacon distance (green) and only heading (red)



achieved by beacon distance and heading. The reason for the drifting is, that feedback of the heading can only correct for errors in the orientation but not for errors in the displacement caused by the bias. For this reason, in practice where the sensor readings both are distorted by orientation and displacement errors, both means of feedback, beacon distance and heading, need to be taken into account.

CONCLUSION AND FUTURE WORK

The differential drive pose update equations have first been written in a closed form expression that simplifies the model implementation as well determining the linearized model for use in the extended Kalman filter, since no checks for equality of the left and right wheel speeds need to be taken anymore. It is also shown that the pose estimation problem is in fact an unknown input estimation problem, which can be solved e.g. by augmenting the state with the unknown input. In this paper, a simplified approach has been taken by first estimating the unknown input resulting in a standard estimation problem, with an additional process noise term. Under some mild assumptions an expression for the covariance of this additional process noise term is derived.

To solve the estimation problem, using measurements from both a distance to a beacon and the readings from a heading sensor, various estimation algorithms are compared: a heuristic sensor fusion algorithm, the extended Kalman filter and the Particle filter.

The simulation experiments show that the extended Kalman filter performs the best, closely followed by the heuristic algorithm. The estimation of the Particle filter algorithm was slightly worse. Simulations based on beacon distance only, neglecting the heading measurements, did not give satisfactory results. Simulations based on heading sensors only, neglecting the beacon distance measurement, showed not to be able to account for systematic errors such as a small bias on one of the wheel speed velocity measurements because it loses information about the displacement error.

Future work is on (automatic) calibration of the parameters in the algorithms and experimental validation on a real differential drive mobile robot.

ACKNOWLEDGMENT

This research was supported by the National Directorate for Practice-oriented Research SIA (Directorate SIA), part of the Netherlands Organization for Scientific Research (NWO) [grant number RAAK.MKB06.014 Let's Move It].

REFERENCES

Dabove, P., Pietra, V. D., Piras, M., Jabbar, A., & Kazim, S. (2018). Indoor positioning using ultra-wide band (UWB) technologies: Positioning accuracies and sensors' performances. *Proceedings of the Position, Location and Navigation Symposium (PLANS)* (pp. 175-184). Piscataway, NJ: IEEE/ION.

Dudek, G., & Jenkin, M. (2010). *Computational Principles of Mobile Robotics*. New York, NY: Cambridge University Press. doi:10.1017/CBO9780511780929

Fraanje, R., Beltman, R., Theinert, F., van Osch, M., Punter, T., & Bolte, J. (2019). Sensor fusion of odometry and a single beacon distance measurement. *Proceedings of the 20th International Conference on Research and Education in Mechatronics (REM 2019)*. Piscataway, NJ: IEEE doi:10.1109/REM.2019.8744098

Gustafsson, F. (2010). *Statistical Sensor Fusion*. Lund, Sweden: Studentlitteratur.

Ohno, K., Tsubouchi, T., Shigematsu, B., & Yuta, S. (2004). Differential GPS and odometry-based outdoor navigation of a mobile robot. *Advanced Robotics*, *18*(6), 611–635. doi:10.1163/1568553041257431

Plets, D., Eryildirim, A., Bastiaens, S., Stevens, M., Martens, L., & Joseph, W. (2017). A performance comparison of different cost functions for RSS-based visible light positioning under the presence of reflections. *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems* (pp. 37-41). New York, NY: ACM. doi:10.1145/3129881.3129888

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA: MIT Press.

Wijk, O., & Christensen, H. (2000). Triangulation-based fusion of sonar data with application in robot pose tracking. *IEEE Transactions on Robotics and Automation*, *16*(6), 740–752. doi:10.1109/70.897785

Rufus Fraanje studied electrical engineering at Delft University of Technology, The Netherlands, and received his M.S. degree in 1999. In 2004 he received his Ph.D. degree at the University Twente, The Netherlands, for his research on robust and fast adaptive control algorithms for noise and vibration control. After several research positions he currently is associate professor in Mechatronics at The Hague University of Applied Sciences, The Netherlands. His research interests are in the areas of intelligent control systems, estimation algorithms, and robotics.

Fidelis Theinert studied electrical engineering and electro-acoustics at the Technische Universitaet Hannover and the Technische Univesitaet Berlin, Germany. He received his MSc degree in 1991. After his research at the Heinrich Hertz Institut, Berlin (Fraunhofer Gesellschaft) he worked in different companies in Germany and The Netherlands as R&D engineer in the field of Embedded Systems and Image Processing. Currently he is working as Senior Lecturer Electrical Engineering and as research fellow in the research group Smart Sensor Systems at The Hague University of Applied Sciences in Delft, The Netherlands.

Michiel van Osch currently is a teacher and researcher at Fontys Applied University in Eindhoven, The Netherlands. He is part of the Mechatronics and Robotics Lectorate of the Fontys department Engineering. His research focusses on the optimization of multi robot and system interaction in a manufacturing environment. Between 2009 and 2015 he was System architect of Robot Rose, a tele-operated service robot for home care and CTO of Rose BV. Michiel van Osch has a PhD. on model-based testing of hybrid systems. These are systems display both discrete and continuous behavior, for instance robots.

Teade Punter is a professor for software development and a lecturer at Fontys University of Applied Sciences in Eindhoven, Netherlands. He has a background in software and systems engineering. He received his master's degree from Twente University in 1991 and his Ph.D. from Eindhoven University of Technology in 2001. Teade's earlier positions were at the Open University of the Netherlands, KEMA Nederland, Fraunhofer IESE. Laboratory for Software Quality at Eindhoven University of Technology, Embedded Systems Institute and TNO-ESI. His research interests are in: software design, modeling, machine learning, integration and test.

John Bolte is full professor and chairs the Smart Sensor Systems group at The Hague University of Applied Sciences. He also works at the National Institute for Public Health and the Environment (RIVM), where he is coordinator for personal exposure and impact monitoring using wearable sensors. He received his MSc in Geophysics from Utrecht University (1995), and his PhD in Applied Physics from Delft University of Technology (2003) for his work on Fresnel zones and focusing operators in acoustical imaging. Since completing his post-initial master's degree at VU University Amsterdam in 2011, he has been a registered epidemiologist. The research lines of his group are: Smart Health: measurement of environmental exposures and effects on humans, animals and plants; Smart Safety: professional exposure and smart and predictive maintenance; and Smart Acquisition and Control: big data analytics, sensor fusion, semantic mapping and autonomous transport.