2019 20th International Conference on Research and Education in Mechatronics (REM)
May 23-24, 2019, Wels, Austria

Sensor fusion of odometry and a single beacon distance measurement

1st Rufus Fraanje
The Hague Univ. of Applied Sciences
Smart Sensor Systems / Mechatronics
Delft, The Netherlands
p.r.fraanje@hhs.nl

2nd René Beltman
Lely
Maassluis, The Netherlands
rbeltman@lely.com

3rd Fidelis Theinert
The Hague Univ. of Applied Sciences
Smart Sensor Systems / Electrical Eng.
Delft, The Netherlands
f.j.theinert@hhs.nl

4th Michiel van Osch
Fontys
Eindhoven, The Netherlands
michiel.vanosch@fontys.nl

5th Teade Punter
Fontys
High Tech Embedded Software
Eindhoven, The Netherlands
teade.punter@fontys.nl

6th John Bolte
The Hague Univ. of Applied Sciences
Smart Sensor Systems
Delft, The Netherlands
j.f.b.bolte@hhs.nl

***Abstract*—The pose estimation of a differential drive robot from noisy odometry and a noisy beacon distance measurement is studied. It is shown that the problem is a state estimation problem with unknown input, which under some assumptions regarding the noise on the state, can be rewritten in a state estimation problem. An heuristic sensor fusion algorithm is proposed and compared with the extended Kalman filter and the particle filter in a simulation experiment.**

***Index Terms*—navigation, odometry, sensor fusion, extended Kalman filter, particle filter**

I. Introduction

In navigation of mobile robots information from multiple sensors need to be used to reliably determine the position of the mobile robot [1]–[3]. In outdoor robotics applications usually GPS or DGPS measurements are used to compensate for the drift in de position estimation based on odometry, see e.g. [4]. For autonomous driving on high-ways and in urban environments also other sensors are being used, such as lidar, radar and/or camera's, see e.g. [1]. For navigation in indoor applications, where GPS or other GNSS has weak or no coverage, other beacons can be used e.g., based on ultrasound transmission time [5] or received signal strength (RSS) of visible light [6] and transmission time of ultra-wideband (UWB) transmission [7]. For indoor navigation also beacon-less methods are being used, that rely on simultaneous localisation and mapping (SLAM) [1] and often make use of optical sensors observing the robots environment, such as lidar and/or (RGBd or stereo) camera's.

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:

This work has been sponsored by the Dutch government under the SIA Raak MKB (SME) regulation.

- *Odometry based:* the position is only determined by odometry;
- *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 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 and beacon based sensors are fused according to some sensor fusion algorithm to provide a position estimate.

In this paper the latter approach of fusing the beacon and odometry sensor measurements is used to achieve a position estimate. 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 and a distance measurements, e.g. determined by UWB. Various sensor fusion algorithms are evaluated, an heuristic approach, the extended Kalman filter and the particle filter, see e.g. [1]–[3]. The algorithms are compared in a simulation experiment.

II. Problem analysis

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

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

where r_ℓ and r_r the wheel radius and $n_{\ell,k}$ and $n_{r,k}$ the number of rotations per second at time index k of the left and right wheel respectively, usually $r_\ell = r_r = r$. The odometry

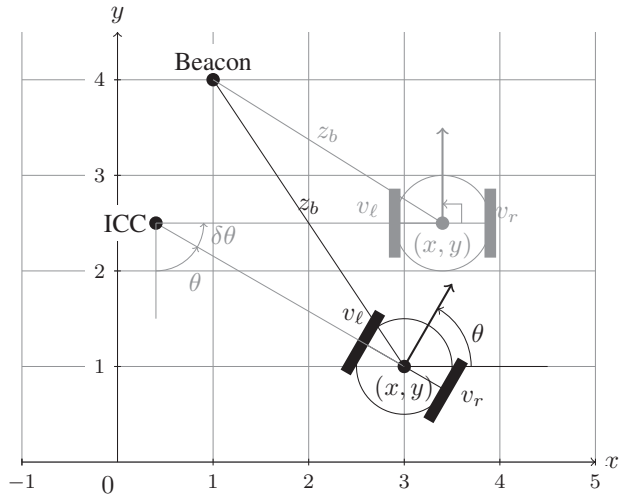

Fig. 1. Schematic of the differential drive mobile robot with the distance to a single beacon being measured. ICC is the Instantaneous Center of Curvature. Dependency on the time index k is suppressed for reason of clarity.

equations that yield the update to the mobile robot pose (position and heading) have been derived in many textbooks, such as [2]. Here, we present an alternative form, that can be derived using some additional trigonometry, which yields

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

where

$$\delta\theta_k = \delta t(v_{r,k} - v_{\ell,k})/b, \tag{3}$$

$$\begin{bmatrix} \delta x_k \\ \delta y_k \end{bmatrix} = s_k \begin{bmatrix} \mathrm{sinc}(\delta\theta_k) & -\sin(\delta\theta_k/2) \\ \sin(\delta\theta_k/2) & \mathrm{sinc}(\delta\theta_k) \end{bmatrix} \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix}, \tag{4}$$

where b the baseline, i.e. the distance between the wheels, and $s_k = \delta t(v_{r,k} + v_{\ell,k})/2$. Note that this expression holds for all $v_{r,k}$, $v_{\ell,k}$ and no condition as $v_{r,k} = v_{\ell,k}$ is needed, as is common in most derivations of the differential drive mobile robot odometry. If $v_{r,k} = v_{\ell,k}$ it follows that $\delta x_k = s_k \cos(\theta_k)$ and $\delta y_k = s_k \sin(\theta_k)$ because $\mathrm{sinc}(0) = 1$. From an implementation point of view as well as in the derivation of the extended Kalman filter this form without conditions is preferred. If $|\delta\theta_k|$ is small ($|\delta\theta_k| \ll 1$), e.g. because the time step is small, $\delta t \ll b/|v_{r,k} - v_{\ell,k}|$, the position updates can be approximated by $\delta x_k = s_k \cos(\theta_k)$ and $\delta y_k = s_k \sin(\theta_k)$. This approximation may sometimes be beneficial from an implementation point of view because it prevents the computation of a $\mathrm{sinc}(\delta\theta_k)$ and $\sin(\delta\theta_k/2)$, but will not be considered in the sequel of this paper.

The measurement of the distance to a beacon, which position is fixed and known, is given by

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

where $\nu_{b,k}$ 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 left and right wheel speeds and the heading are measured, e.g. by means of wheel encoders or tachometers and a compass or gyroscope:

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

where $\nu_{vr,k}$, $\nu_{v\ell,k}$ and $\nu_{\theta,k}$ represent measurement noises.

For notational convenience the following quantities are introduced. The pose is denoted by

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

The command vector

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

is directly determined by the left and right wheel velocities $v_{\ell,k}$ and $v_{r,k}$:

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

Then the pose update equation is written by

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

where the function $f(.)$ can be easily derived from (2):

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

where $u(i)$ refers to the i^{th} element of the vector u etc., and the dependency on the time index k is suppressed for clarity.

The measurements are stored in one measurement vector

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

and can be written as

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

where ν_k the vector with the measurement noises and the function $g(.)$ defined as

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

Then, the problem is to (recursively) estimate the pose q_k given the noisy measurements z_k. Note, that this is an estimation problem with an *unknown* input u_k, because the measured wheel velocities, $z_k(2)$ and $z_k(3)$, are distorted by noise from which only a noise deteriorated estimate of u_k can be determined!

To simplify the problem, we slightly reformulate the problem, by introducing the command vector determined from measurements:

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

Then the pose update equation is written in terms of $\hat{u}_k$ rather than u_k as

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

where $\eta_k = f(q_k, u_k) - f(q_k, \hat{u}_k)$ is a noise term to take into account the error in using $\hat{u}_k$ rather than u_k.

Because the measurements $z_k(2)$ and $z_k(3)$ have been used already in determining $\hat{u}_k$ they are not considered as outputs and we introduce the reduced output consisting of the beacon distance and the heading:

$$z'_k \;=\; \begin{bmatrix} z_k(1) & z_k(4) \end{bmatrix}^T \;=\; g'(q_k) + \nu'_k, \tag{17}$$

where

$$g'(q) \;=\; \begin{bmatrix} \sqrt{(x_b - q(1))^2 + (y_b - q(2))^2} \\ q(3) \end{bmatrix}, \tag{18}$$

and $\nu'_k = \begin{bmatrix} \nu_k(1) & \nu_k(4) \end{bmatrix}^T$. With these adjustments, there is a noise term η_k introduced in the pose update equation but the unknown input in the estimation problem is removed. Note, that η_k cannot be considered as independent of u_k and q_k, which however will be assumed e.g. in application of the extended Kalman filter below.

III. Sensor fusion algorithms

A. 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 than to fuse this estimate with the measurement from the distance to the beacon.

- Set the initial pose estimate $\hat{q}_{0|0}$ and the initial command vector estimate $\hat{u}_0$;
- Iterate for $k = 1, 2, \cdots$:
  1) Perform a time update of the pose estimate:
  $$\hat{q}_{k|k-1} \;=\; f(\hat{q}_{k-1|k-1}, \hat{u}_{k-1}). \tag{19}$$
  2) Measure z_k and determine z'_k and $\hat{u}_k$.
  3) Determine the predicted output:
  $$\hat{z}'_k \;=\; g'(\hat{q}_{k|k-1}). \tag{20}$$
  4) Make a weighted average of the heading (fusion step):
  $$\hat{q}_{k|k}(3) \;=\; \frac{C^{-1}_{\theta,p}\hat{z}'_k(2) + C^{-1}_{\theta,m}z'_k(2)}{C^{-1}_{\theta,p} + C^{-1}_{\theta,m}}, \tag{21}$$
  where $C_{\theta,p}$ the variance of the error in the heading estimate $\hat{z}'_k(2)$ and $C_{\theta,m}$ the variance of the noise in measurement $z'_k(2)$.
  5) Update the position by fusing the beacon distance measurement:
     a) Calculate the angular position estimate in the beacons coordinate frame:
     $$\hat{\phi}_{k|k-1} = \text{atan2}(\hat{q}_{k|k-1}(2) - y_b, \hat{q}_{k|k-1}(1) - x_b). \tag{22}$$
     b) Make a weighted average of the distance to the beacon (fusion step):
     $$\hat{z}'_{k|k}(1) \;=\; \frac{C^{-1}_{b,p}\hat{z}'_k(1) + C^{-1}_{b,m}z'_k(1)}{C^{-1}_{b,p} + C^{-1}_{b,m}}, \tag{23}$$
     where $C_{b,p}$ the variance of the error in the distance to the beacon estimate $\hat{z}'_k(1)$ and $C_{b,m}$ the variance of the noise in measurement $z'_k(1)$.
     c) Calculate the position estimate:
     $$\begin{bmatrix} \hat{q}_{k|k}(1) \\ \hat{q}_{k|k}(2) \end{bmatrix} \;=\; \begin{bmatrix} x_b \\ y_b \end{bmatrix} + \hat{z}_{k|k}(1) \begin{bmatrix} \cos(\hat{\phi}_{k|k-1}) \\ \sin(\hat{\phi}_{k|k-1}) \end{bmatrix}. \tag{24}$$

The algorithm heavily relies on the fusion equation of two estimates each with its own error variance, see e.g. [3, Sec. 2.3.1]. The implementation is rather straightforward, but the difficulty in practical use is to determine the variances $C_{\theta,p}$, $C_{\theta,m}$, $C_{b,p}$ and $C_{b,m}$.

B. Extended Kalman filter

The extended Kalman filter is obtained by applying the Kalman filter to the linearised pose update and measurement equation, i.e. the functions $f(.)$ and $g'(.)$:

$$A \;=\; \frac{\partial f}{\partial q} \;=\; \begin{bmatrix} 1 & 0 & -u(2)(\mathrm{s}_{12}\mathrm{c}_{q3} + \mathrm{s}_{\mathrm{c}1}\mathrm{s}_{q3}) \\ 0 & 1 & u(2)(\mathrm{s}_{\mathrm{c}1}\mathrm{c}_{q3} - \mathrm{s}_{12}\mathrm{s}_{q3}) \\ 0 & 0 & 1 \end{bmatrix}, \tag{25}$$

$$B \;=\; \frac{\partial f}{\partial u} \;=\; \begin{bmatrix} u(2)(\alpha\mathrm{c}_{q3} - \mathrm{c}_{12}\mathrm{s}_{q3}/2) & \mathrm{s}_{\mathrm{c}1}\mathrm{c}_{q3} - \mathrm{s}_{12}\mathrm{s}_{q3} \\ u(2)(\mathrm{c}_{12}\mathrm{c}_{q3}/2 + \alpha\mathrm{s}_{q3}) & \mathrm{s}_{12}\mathrm{c}_{q3} + \mathrm{s}_{\mathrm{c}1}\mathrm{s}_{q3} \\ 1 & 0 \end{bmatrix}, \tag{26}$$

where

$$\mathrm{s}_{\mathrm{c}1} = \mathrm{sinc}(u(1)), \qquad \mathrm{s}_{12} = \sin(u(1)/2), \tag{27}$$

$$\mathrm{c}_{12} = \cos(u(1)/2), \quad \mathrm{s}_{q3} = \sin(q(3)), \tag{28}$$

$$\mathrm{c}_{q3} = \cos(q(3)), \qquad \alpha = \frac{u(1)\cos(u(1)) - \sin(u(1))}{u^2(1)}, \tag{29}$$

and

$$C \;=\; \frac{\partial g'}{\partial q} \;=\; \begin{bmatrix} \frac{-(x_b - q(1))}{z_b} & \frac{-(y_b - q(2))}{z_b} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{30}$$

where

$$z_b \;=\; \sqrt{(x_b - q(1))^2 + (y_b - q(2))^2}. \tag{31}$$

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

$$q_{k+1} = A_k q_k + B_k \hat{u}_k + \eta'_k, \tag{32}$$

$$z'_k = C_k q_k + \nu'_k. \tag{33}$$

Assume that the disturbances η'_k and ν'_k can be approximated by independent wide sense stationary stochastic white noise processes with correlation

$$\mathrm{E}\left(\begin{bmatrix} \eta'_k \\ \nu'_k \end{bmatrix} \begin{bmatrix} \eta'_i \\ \nu'_i \end{bmatrix}^T \right) = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \delta(k - i), \tag{34}$$

where $\delta(k)$ the Kronecker delta function, i.e. $\delta(0) = 1$ and $\delta(k) = 0$ for $k \neq 0$. This is a rather strong assumption. It may be relaxed by allowing the covariance matrices Q and R to be depending on the time index k. Because it is far from straightforward to determine the covariance matrices for each k we assume they can be considered constant.

The estimation of extended Kalman filter is given (e.g. [3]):

- 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 $\hat{u}_0$.
- Iterate for $k = 1, 2, \cdots$:
  1) Calculate A_{k-1} and B_{k-1} using $\hat{q}_{k-1|k-1}$ and $\hat{u}_{k-1}$
  2) Perform a time update of the pose estimate:
  $$\hat{q}_{k|k-1} = f(\hat{q}_{k-1|k-1}, \hat{u}_{k-1}), \quad (35)$$
  and the pose prediction error covariance:
  $$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^T + Q. \quad (36)$$
  3) Calculate C_k using $\hat{q}_{k|k-1}$.
  4) Measure z_k and determine z'_k and $\hat{u}_k$.
  5) Determine the predicted output:
  $$\hat{z}'_k = g'(\hat{q}_{k|k-1}), \quad (37)$$
  and the innovation:
  $$e_k = z'_k - \hat{z}'_k, \quad (38)$$
  and the Kalman gain:
  $$K_k = P_{k|k-1}C_k^T(C_kP_{k|k-1}C_k^T + R)^{-1}. \quad (39)$$
  6) Perform a measurement update of the pose estimate:
  $$\hat{q}_{k|k} = \hat{q}_{k|k-1} + K_ke_k, \quad (40)$$
  and the pose estimation error covariance:
  $$P_{k|k} = P_{k|k-1} - K_kC_kP_{k|k-1}. \quad (41)$$

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 linearization is only used in calculating the pose prediction and estimation covariances and the Kalman gain.

C. 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. [3], relaxes this assumption, by sampling the probability density function by a number of so called particles 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. Therefore, the probability density function of the measurement noise μ' is needed, which is assumed to be zero mean and Gaussian with constant covariance $R_{\nu'}$, i.e.,

$$p_{\nu'}(\xi) = \frac{e^{-\frac{1}{2}\xi^T R_{\nu'}\xi}}{2\pi\sqrt{|R_{\nu'}|}}. \quad (42)$$

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

- Choose (different) initial values for the N_p particles $\{\hat{q}^i_{0|0}\}_{i=1}^{N_p}$.
- Iterate for $k = 1, 2, \cdots$:
  1) Perform a time update of the pose estimate particles:
  $$\hat{q}^i_{k|k-1} = f(\hat{q}^i_{k-1|k-1}, \hat{u}_{k-1}), \quad i = 1, \cdots, N_p. \quad (43)$$
  2) Measure z_k and determine z'_k and $\hat{u}_k$
  3) Determine for each particle the predicted output:
  $$\hat{z}'^i_k = g'(\hat{q}^i_{k|k-1}), \quad i = 1, \cdots, N_p. \quad (44)$$
  4) Calculate the weights with the probability density function of the noise on z'_k:
  $$w^i_{k|k} = \frac{p_{\nu'}(z'_k - \hat{z}'^i_k)}{c_k}, \quad i = 1, \cdots, N_p, \quad (45)$$
  where $c_k = \sum_{i=1}^{N_p} p_{\nu'}(z'_k - \hat{z}'^i_k)$ the normalization weight.
  5) Calculate the pose estimate by
  $$\hat{q}_{k|k} = \sum_{i=1}^{N_p} w^i_{k|k}\hat{q}^i_{k|k-1}. \quad (46)$$
  6) Take samples $\{\hat{q}^i_{k|k}\}_{i=1}^{N_p}$ with replacement from the set $\{\hat{q}^i_{k|k-1}\}_{i=1}^{N_p}$ where the probability to take sample i is $w^i_{k|k}$.
  7) To prevent particle depletion (because of resampling with replacement more and more particles may become equal), a small amount of noise is added:
  $$\hat{q}^i_{k|k-1} \leftarrow \hat{q}^i_{k|k-1} + \zeta_k, \quad (47)$$
  where ζ_k is a realization of a zero mean (Gaussian) random vector with covariance ϵQ where ϵ a small number (e.g. $\epsilon = 10^{-4}$).

IV. Simulation experiment

The algorithms of the previous section are compared in a simulation experiment of a differential drive mobile robot with baseline $r_l = r_r = 0.5$ m and the time update rate is $\delta t = 0.01$ s. The first 100 samples the left and right wheel velocity are both 1 m/s, then for 200 samples the left wheel velocity is 0.56 m/s and the right is 0.95 m/s, followed by 200 samples where the left wheel velocity is 0.95 m/s and the right is 0.56 m/s. The last 99 samples the wheel velocities are both again 1 m/s. The starting position of the robot is the origin, i.e. $(0, 0)$ m with heading 0^o. The beacon is at position $(1, 1)$ 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 0.001 m^2, of the right and left wheel velocity measurement is 0.001 m^2/s^2 and of the heading measurement is 0.001 $(^o)^2$.

In practice it is impossible to exactly determine the wheel's radius. To simulate the effect of a small error in one of

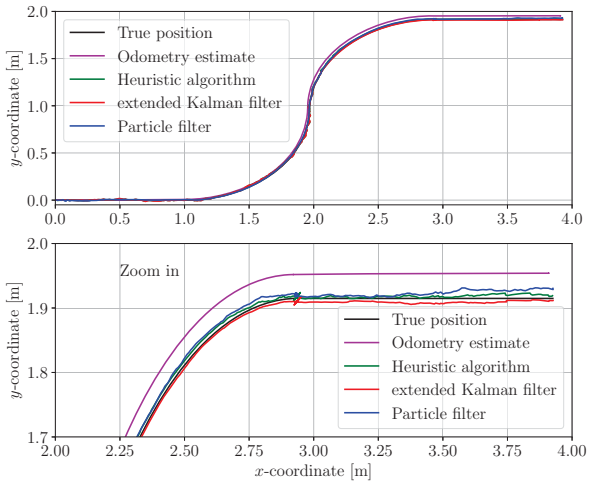

Fig. 2. Example trajectory (black) and position estimates obtained by odometry (pink), heuristic algorithm (green), extended Kalman filter (red) and the particle filter (blue).

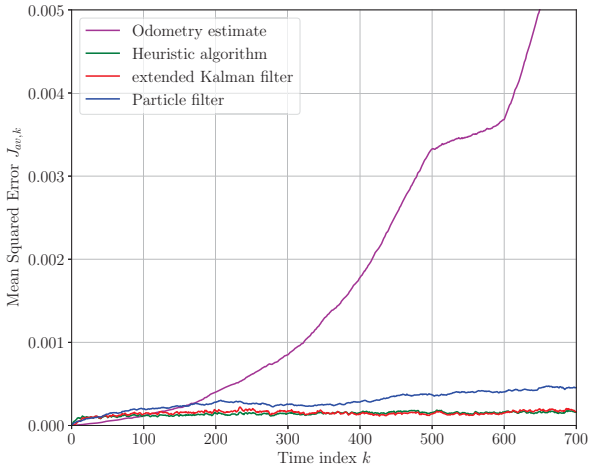

Fig. 3. Mean square error of the pose averaged over $N_{exp} = 100$ simulations versus time index k obtained by odometry, the heuristic algorithm, extended Kalman filter and the particle filter; beacon distance and heading are used in the estimation.

the wheel radii we add a small bias of 0.002 m/s on the measurement of the left wheel velocity.

For the heuristic approach the variances of the noise on the beacon distance measurement and the heading measurement are chosen to be the same as the variances of the imposed measurement noises, i.e. $C_{b,m} = 0.001$ m^2 and $C_{\theta,m} = 0.001$ $(^o)^2$ respectively. The variances of the prediction errors are set to a factor 10 smaller, i.e. $C_{b,p} = 0.0001$ m^2 and $C_{\theta,p} = 0.0001$ $(^o)^2$, to give more weight to the model based prediction, which turned out to give a smoother estimate.

For the extended Kalman filter, the $P_{0|0}$ and Q covariance matrices are determined by a bit of trial and error and set to $P_{0|0} = 10^{-5} I_3$ and $Q = 10^{-5} I_3$. The covariance of the measurement noise is determined by the variances of the measurement noises, i.e. $R_{\nu'} = \text{diag}(0.001, 0.001)$.

For the particle filter $N_p = 100$ particles have been used and the scaling parameter of the spurious noise on the particle to prevent particle depletion was chosen as $\epsilon = 0.1$.

Fig. 2 shows the true trajectory (position only) of the mobile robot (black) as well as the trajectory achieved by just odometry (pink), the heuristic algorithm (green), the extended Kalman filter (red) and the particle filter (blue). From the zoom in sub figure it is clear that the odometry yields a significant deviation, which is due to the offset in the wheel speed measurement. The estimation algorithms (heuristic algorithm, extended Kalman and particle filter) provide reasonably well estimates.

To study the average behaviour, the simulation has been repeated over $N_{exp} = 100$ experiments, and the mean squared error (MSE) of the pose estimation is calculated for each k:

$$J_{av,k} = \frac{1}{N_{exp}} \sum_{j=1}^{N_{exp}} \| [q_k - \hat{q}_{k|k}]_j \|_2^2, \qquad (48)$$

where $[q_k - \hat{q}_{k|k}]_j$ denotes the estimation error of the j-th experiment. Fig. 3 shows the evolution of $J_{av,k}$ over time for the odometry, the heuristic algorithm, the extended Kalman filter and the particle filter. Note, that both the measured distance to the beacon and the heading are used in the estimation, as described in the previous section. Because of the error in the wheel velocity measurements, for which the odometry does not compensate, the mean squared error diverges significantly. The heuristic algorithm as well as the extended Kalman filter show rather accurate estimates. It was surprising that the heuristic algorithm gives a rather good performance. The accuracy of the particle filter is slightly worse, which might be due to the noise that we added to prevent particle depletion (i.e. $\epsilon \neq 0$).

To get insight how important the individual use of the distance measurement and the heading are in estimating the pose, similar simulations have been performed under the condition that only the beacon distance measurement has been used, and also only the heading measurement. Fig. 4, 5 and 6 show the mean squared error $J_{av,k}$ under the different measurement conditions for the heuristic algorithm, the extended Kalman filter and the particle filter respectively.

For all algorithms disregarding the heading information resulted in a diverging $J_{av,k}$, so including heading information in the estimation is most important in this simulation experiment. For the heuristic algorithm including the measurement of the distance to the beacon did not have a significant influence on the performance. For the extended Kalman filter and the particle filter including the beacon distance measurement, in addition to the heading measurement, resulted in a notable improvement in the pose estimation. For the particle filter not including the beacon distance measurement resulted even in a diverging mean squared error.

Note, that in this simulation experiment the extended

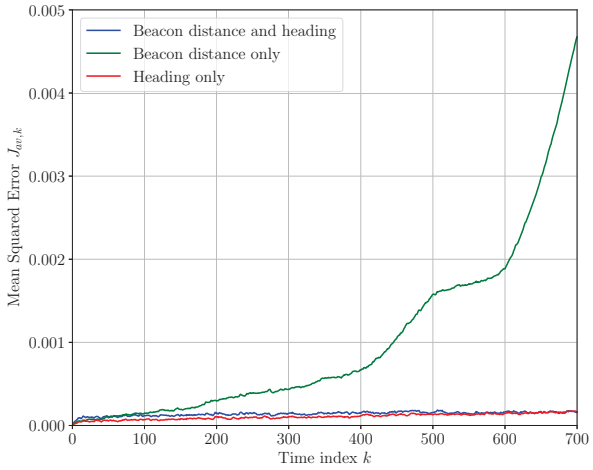

Fig. 4. Mean square error of the pose averaged over $N_{exp} = 100$ simulations versus time index k obtained by the heuristic algorithm using both beacon distance and heading, only beacon distance and only heading.

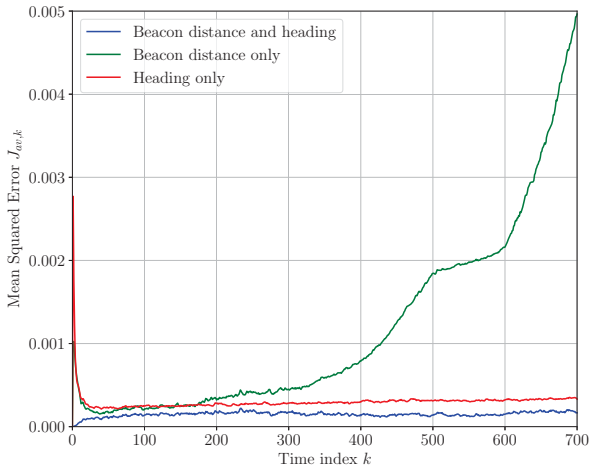

Fig. 5. Mean square error of the pose averaged over $N_{exp} = 100$ simulations versus time index k obtained by the extended Kalman filter using both beacon distance and heading, only beacon distance and only heading.

Kalman filter gave the best results. But note, that the simulation experiment is rather limited, and only evaluated for a specific path.

V. Conclusions and Future Work

The differential drive odometry is rewritten in an explicit form, without conditions, which is beneficial in implementation as well as in deriving the extended Kalman filter. By assuming that the noise wheel speed measurements can be considered as inputs to the system, the pose estimation problem is reformulated in a state estimation problem with known inputs. A heuristic sensor fusion algorithm is proposed

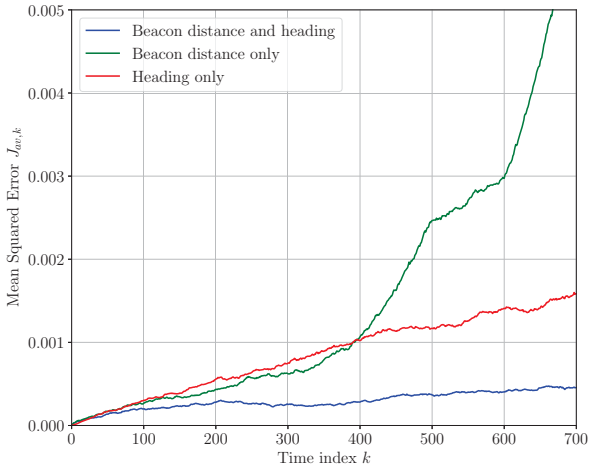

Fig. 6. Mean square error of the pose averaged over $N_{exp} = 100$ simulations versus time index k obtained by the particle filter using both beacon distance and heading, only beacon distance and only heading.

and compared with the extended Kalman filter and the particle filter. The initial simulation experiments show that all three algorithms perform quite well. It turns out that feedback from the heading sensor is important, especially at making turns. Also the feedback from the distance to the beacon may improve the accuracy, as was demonstrated for the extended Kalman filter and the particle filter.

Future work is on (automatic) calibration, determination of the covariance matrices, extensive simulation experiments and experimental validation on a differential drive mobile robot.

Acknowledgement

Thanks to the (former) students Jasper Brink, Quinty van Dijk, Tom Hoefnagel, Martijn Keijzer, Dennis Morits, Kevin Schomper, Nick Visser, Rames Yanes and Chadier Wilson, who have contributed to parts of this paper.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
[2] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010.
[3] F. Gustafsson, *Statistical Sensor Fusion*. Studentlitteratur, 2010.
[4] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta, "Differential GPS and odometry-based outdoor navigation of a mobile robot," *Advanced Robotics*, vol. 18, no. 6, pp. 611–635, 2004.
[5] O. Wijk and H. Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking," *IEEE Trans. Robotics and Automation*, vol. 16, no. 6, pp. 740–752, 2000.
[6] D. Plets, A. Eryildirim, S. Bastiaens, M. Stevens, L. Martens, and W. Joseph, "A performance comparison of different cost functions for rss-based visible light positioning under the presence of reflections," in *Proceedings of the 4th ACM Workshop on Visible Light Communication Systems*, pp. 37–41, 2017.
[7] P. Dabove, V. D. Pietra, M. Piras, A. Jabbar, and S. Kazim, "Indoor positioning using ultra-wide band (uwb) technologies: Positioning accuracies and sensors' performances," in *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION*, pp. 175–184, 2018.