Mobile Robot Control Using a Cloud of Particles ⋆

Jorge Augusto Vasconcelos Alves * **Walter Fetter Lages** *

* *Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil (e-mail: {jorge, fetter}@ece.ufrgs.br)*

Abstract: Common control systems for mobile robots include the use of deterministic control laws together with state estimation approaches and the consideration of the certainty equivalence principle. Recent approaches consider the use of partially observable Markov decision process strategies together with Bayesian estimators. In order to reduce the required processing power and yet allow for multimodal or non-Gaussian distributions, a scheme based on a particle filter and a corresponding cloud of input signals is proposed in this paper. Results are presented and compared to a scheme with extended Kalman filter and the assumption that the certainty equivalence holds.

Keywords: Mobile robotics, particle filter, nonlinear control, stochastic control, stochastic estimation.

1. INTRODUCTION

Mobile robots are known to be subject to uncertainties in both the robot behavior and the environment where the robot navigates (Thrun et al., 2005). The classic approach for state estimation and control of stochastic systems is to consider the expected value of the system state variables and the certainty-equivalence principle (Anderson and Moore, 1989). Expected value approaches, however, cannot be used when multimodal (or even skewed) distributions are present. On the other hand, skewed or multimodal distributions can arise due to sensor fusion and typical mobile robotics problems (Kaelbling et al., 1998). In addition to it, nonlinear dynamics often generate multimodal or skewed distributions from normal or uniform distributions. The current state-of-the-art approach to cope with uncertainties, specially those with non-Gaussian probability distributions, is to use Bayesian filters to estimate the system state and then obtain a control signal based on the result of the estimation, which is either a probability density, a histogram, a set of particles or probabilities over a topological map. This signal can be obtained from a mode or through optimization, such as partially observable Markov decision processes (POMDP) approaches (Blanco et al., 2010; Thrun et al., 2005). The use of POMDP for systems with continuous states demands approximations, or the problem becomes intractable (Baral et al., 2000; Littman et al., 1995).

This paper proposes a control scheme for a differential-drive mobile robot that maps a set of possible states into a space of control signals. Both the state transition and observations are subject to uncertainties. Hence, a particle filter is proposed for state estimation. However, the state estimation is not taken to be a single vector, but the full cloud of particles which represents the probability of each state vector. Then, a globally stable control law is considered for the mapping of the cloud of particles in state-space into a cloud of particles in input-space. The input signal to be applied to the robot is then chosen among those in the most populated regions in the input space. In addition to it, the observations are restricted in sampling frequency, so that an absolute reference, which was assumed to be a GPS, is only available from time to time, while encoder measurements are obtained at every sampling instant. Finally, a comparison with classic control approaches is presented.

2. SYSTEM DESCRIPTION

The kinematic motion of differential-drive mobile robots, moving on an horizontal plane, is described in continuous time by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \cos x_3 & 0 \\ \sin x_3 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \tag{1}$$

where $\mathbf{x} = [x_1 \ x_2 \ x_3]^{\mathrm{T}}$ is the state vector and $\mathbf{u} = [u_1 \ u_2]^{\mathrm{T}}$ is the input vector. The state variables x_1 and x_2 are the plane coordinates, x_3 is the orientation angle, and the input variables u_1 and u_2 are the linear and angular speeds, respectively.

A discrete-time realization (Lages, 1998) is given by

$$\begin{aligned} \mathbf{x}(k+1) &= f_d\left(\mathbf{x}(k), \mathbf{u}(k)\right) \\ &= \mathbf{x}(k) + \begin{bmatrix} Tu_1(k)\left[\operatorname{sinc}\left(\dfrac{Tu_2(k)}{2}\right)\cos\left(x_3(k) + \dfrac{Tu_2(k)}{2}\right)\right] \\ Tu_1(k)\left[\operatorname{sinc}\left(\dfrac{Tu_2(k)}{2}\right)\sin\left(x_3(k) + \dfrac{Tu_2(k)}{2}\right)\right] \\ Tu_2(k) \end{bmatrix} \end{aligned} \tag{2}$$

where $\operatorname{sinc} x \triangleq \frac{\sin x}{x}$ and T is the sampling period.

The robot behavior can be better described by stochastic models, as presented by Thrun et al. (2005) and Rekleitis

⋆ The authors would like to thank the financial support from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and Fundação de Apoio à Pesquisa do Estado do Rio Grande do Sul (FAPERGS).

(2004). These account for two types of errors: systematic errors and non-systematic errors. Systematic errors can be compensated for by appropriate calibration of kinematic parameters (Borenstein et al., 1996; Rekleitis, 2004). The non-systematic effects are due to stochastic effects and can not be compensated by calibration.

The stochastic effects can be observed in the robot motion by a drift of the robot with respect to the nominal trajectory in both traveled distance and orientation. As drifts, those errors increase with time, therefore, they will be modeled here as uncertain linear and angular speeds of the robot. Furthermore, the stochastic effects were found to be closely related to the linear speed of the model (Rekleitis, 2004). Hence, the stochastic version of (1) is

$$\dot{\mathbf{x}}(t) = f\left(\mathbf{x}(t), \mathbf{u}(t) + \mathbf{w}(t)\right), \tag{3}$$

with $\mathbf{w}(t) = u_1(t)[w_t(t)\ w_D(t)]^{\mathrm{T}}$, where $w_t(t) \sim \mathcal{N}\left(0, \sigma_t^2\right)$ and $w_D(t) \sim \mathcal{N}\left(0, \sigma_D^2\right)$ are Gaussian processes representing the uncertainty in linear and angular speeds, respectively.

In order to obtain a discrete model that can properly represent the orientation uncertainty at $k+1$, it will be assumed here, with basis on Rekleitis (2004), that half of the effects of the uncertain angular speed acts through the state transition, therefore affecting both position and orientation at $k+1$, and the other half acts directly on the orientation at $k+1$. Hence, the uncertainty $w_D(t)$ in the continuous model will be represented by two uncertainties in the discrete model: $w_{d_1}(k) \sim \mathcal{N}\left(0, \sigma_d^2\right)$, which acts though the state transition and $w_{d_2}(k) \sim \mathcal{N}\left(0, \sigma_d^2\right)$, which acts directly on state at $k+1$. The effects of $w_t(t)$ can be directly mapped in $w_t(k) \sim \mathcal{N}\left(0, \sigma_t^2\right)$. Then, discrete model can be written as:

$$\mathbf{x}(k+1) = f_d\left(\mathbf{x}(k), \mathbf{u}(k) + \mathbf{w}_1(k)\right) + \mathbf{w}_2(k), \tag{4}$$

where the state transition $f_d(\,\cdot\,,\,\cdot\,)$ is given by (2) and

$$\mathbf{w}_1(k) \triangleq u_1(k)\begin{bmatrix} w_t(k) \\ w_{d_1} k) \end{bmatrix},$$

$$\mathbf{w}_2(k) \triangleq T u_1(k)\begin{bmatrix} 0 \\ 0 \\ w_{d_2}(k) \end{bmatrix}$$

Since $w_{d_1}(k)$ and $w_{d_2}(k)$ are assumed to represent half of the effects of $w_D(t)$, their variance should be half of σ_D^2, or

$$\sigma_d = \frac{\sigma_D}{\sqrt{2}}.$$

It must be noted that while one reads $\mathbf{w}_1(k)$, $\mathbf{w}_1(k)$ and $\mathbf{w}_2(k)$ as addends in (3) and (4), they actually are not additive uncertainties, since they depend on the linear speed $u_1(k)$ and both $f(\,\cdot\,,\,\cdot\,)$ and $f_d(\,\cdot\,,\,\cdot\,)$ are nonlinear.

The model (4) will be used for estimating the state transitions for a set of possible values for the state vector, which will be explained in detail in section 3.1. It is important to note that even though $w_t(k)$, $w_{d_1}(k)$ and $w_{d_2}(k)$ are assumed Gaussian, the resulting state $\mathbf{x}(k+1)$ is not Gaussian, due to nonlinearities. The model (3) will be used for simulating the robot in section 4 while model (4) will be used for state estimation.

3. CONTROL SYSTEM

3.1 Estimation

Most mobile robots can not accurately and timely determine their pose (i.e., position and orientation) from sensor observations. Here, we assume that the robot is equipped with optical encoders on the wheels and a GPS, only. The encoder readings are considered a measurement from the inputs, while the GPS provides measurement of the pose. However, while the encoder observations are available at every sampling instant and assumed accurate for short distances, the GPS observations are noisy and only available at a longer sampling period. On the other hand, the GPS measurements is key in order to correct the drift error which results from dead reckoning and increases with time.

Since encoder readings are mapped internally as a measurement of the input $\mathbf{u}(k)$ the observation vector $\mathbf{y}(k)$, which is corrupted by noise $\mathbf{v}(k)$, is restricted to the GPS observation

$$\mathbf{y}(k) = h\left(\mathbf{x}(k), \mathbf{v}(k)\right) = \mathbf{x}(k) + \mathbf{v}(k). \tag{5}$$

The particle filter, however, could be extended to consider more and other types of sensors, by just considering them in the definition of $h\left(\mathbf{x}(k), \mathbf{v}(k)\right)$.

Note that for other types of sensors the mapping from $\mathbf{x}(k)$ and $\mathbf{v}(k)$ to $\mathbf{y}(k)$ can be nonlinear and that when redundant sensors are used, the dimension of $\mathbf{y}(k)$ can be greater than that of $\mathbf{x}(k)$.

As is the case for many Bayesian filter techniques, particle filter algorithms can be broken into two different stages, called *prediction* and *update*. The particle filter scheme is summarized in the following.

At each sampling instant, possible values for the state vector $\mathbf{x}_i(k)$, $i \in [1, M]$, are considered, based on the previous observations from the system. Each vector $\mathbf{x}_i(k)$ is called a *particle* and M is the total number of particles. The *state belief* $\mathrm{bel}_p\left(\mathbf{x}(k)\right)$ is given by the set of all such particles, i.e.,

$$\mathrm{bel}_p\left(\mathbf{x}(k)\right) = \{\mathbf{x}_1(k), \mathbf{x}_2(k), \ldots, \mathbf{x}_M(k)\}. \tag{6}$$

The state belief is an approximation of a probability density function in the following sense: state space regions with a relatively large number of particles have high probability density values, while regions with relatively few particles are supposed to have low density values.

The prediction step of the algorithm takes the state belief and the system input vector as arguments to generate the *prior* state belief $\overline{\mathrm{bel}}_p\left(\mathbf{x}(k+1)\right)$. For each particle, a new one is generated, according to the state transition function of the system (4), being that the uncertain terms are obtained from pseudo-random number generators with the appropriate distributions. We note that this can be done for any distribution. The *prior* state belief is noted as

$$\overline{\mathrm{bel}}_p\left(\mathbf{x}(k+1)\right) = \{\mathbf{x}_1(k+1), \mathbf{x}_2(k+1), \ldots, \mathbf{x}_M(k+1)\}, \tag{7}$$

where each particle $\mathbf{x}_i(k+1)$, $i \in [1, M]$, is obtained as

$$\mathbf{x}_i(k+1) = f_d\left(\mathbf{x}_i(k), \mathbf{u}(k) + \mathbf{w}_{1_i}(k)\right) + \mathbf{w}_{2_i}(k). \tag{8}$$

The set of particles $\overline{\mathrm{bel}}_p\left(\mathbf{x}(k+1)\right)$ is obtained without information from the system observation at $k+1$: it only takes the set of particles $\mathrm{bel}_p\left(\mathbf{x}(k)\right)$ and the input signal as arguments. This set of particles is then updated with the information from the observations, returning the state belief $\mathrm{bel}_p\left(\mathbf{x}(k+1)\right)$ at $k+1$. This is accomplished by obtaining a so-called *importance factor* $\iota_i(k+1)$ for each particle $\mathbf{x}_i(k+1)$ according to

$$\iota_i(k) = f_y\left(\mathbf{y}(k)|\mathbf{x}_i(k)\right),$$

with

$$f_y\left(\mathbf{y}(k)|\mathbf{x}(k)\right) = \frac{e^{\left(-\frac{1}{2}[\mathbf{y}(k)-\mathbf{C}\mathbf{x}(k)]^{\mathrm{T}}\mathbf{P}[\mathbf{y}(k)-\mathbf{C}\mathbf{x}(k)]\right)}}{\sqrt{(2\pi)^n|\mathbf{P}|}},$$

where $\mathbf{P}$ is the noise covariance matrix and n is the number of rows of $\mathbf{y}(k)$. The state belief $\mathrm{bel}_p\left(\mathbf{x}(k+1)\right)$ is obtained by selecting particles, among those in $\overline{\mathrm{bel}}_p\left(\mathbf{x}(k+1)\right)$, with a probability which is proportional to its importance factor, as presented by Thrun et al. (2005).

The belief update demands an observation to take place. In this paper, a GPS was considered to have a sampling period lower than that of the incremental encoders. As a result, the update step does not occur at every sampling instant, but only when the observation from the GPS is obtained. A more detailed presentation of the particle filter is available in Thrun et al. (2005).

3.2 Control

Differential-drive mobile robots are non-holonomic systems (Campion et al., 1996). An important general statement on the control of non-holonomic systems has been made by Brockett (1982), who has shown that it is not possible to asymptotically stabilize the system at an arbitrary point through a time-invariant, smooth state feedback law. In spite of it, the system is controllable (Astolfi, 1994).

In this paper, we will obtain a set of possible input signals based on non-smooth control law which is obtained by a non-smooth coordinate transformation. A general way of designing control laws for non-holonomic systems through non-smooth coordinate transformations was presented by Astolfi (1994). We have considered a mapping from the state space to the input space as presented by Lages and Hemerly (1998).

The mappings from the system state to the input space which are used for point stabilization are such that the state space origin is made asymptotically stable. If we represent the mapping as $g: \mathbf{X} \to \mathbf{U}$, $\mathbf{x} \in \mathbf{X}$ and $\mathbf{u} \in \mathbf{U}$, then the autonomous system

$$\dot{\mathbf{x}} = f\left(\mathbf{x}, g(\mathbf{x})\right)$$

where $f(\,\cdot\,,\cdot\,)$ is described by (1), is asymptotically stable at the origin. However, it is of interest to stabilize the robot at any point $\mathbf{x}_r$, which means any given position and orientation $(x_{r_1}, x_{r_2}, x_{r_3})$. This can be accomplished by the coordinate change $\bar{\mathbf{x}}(\mathbf{x}, \mathbf{x}_r)$, obtained by the transformation

$$\bar{\mathbf{x}} = \begin{bmatrix} \mathbf{R}(x_{r_3}) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} (\mathbf{x} - \mathbf{x}_r) \tag{9}$$

where $\mathbf{R}(x_{r_3})$ is a 2-D rotation matrix, that is,

$$\mathbf{R}(x_{r_3}) = \begin{bmatrix} \cos x_{r_3} & \sin x_{r_3} \\ -\sin x_{r_3} & \cos x_{r_3} \end{bmatrix}.$$

Hence, if the system $\dot{\bar{\mathbf{x}}} = f\left(\bar{\mathbf{x}}, g(\bar{\mathbf{x}})\right)$ is stable at $\bar{\mathbf{x}} = \mathbf{0}$, then $\dot{\mathbf{x}} = f\left(\mathbf{x}, g(\mathbf{x})\right)$ is stable at $\mathbf{x} = \mathbf{x}_r$. Therefore, in order to stabilize the system at any arbitrary point $\mathbf{x}_r$ based on a control law g that leads the state to the origin, it suffices to use $g(\bar{\mathbf{x}})$ (Sørdalen, 1993).

Low level mobile robot control schemes usually take the state vector as input. However, here, the estimation result is a set of particles. This resulting estimation may have points grouped around different regions, as a result of multimodal beliefs. As a consequence, either a mean squared estimation or the expected value are not appropriate estimation results, and the certainty equivalence principle can not be applied. We present a way of generating a control signal from the current belief by considering the resulting signal from each of the belief particles, and verifying their distribution in the space of the inputs. This way, not only an appropriate action can be found, but it can be reasoned whether an action is appropriate at a given instant, depending on the resulting set of control particles.

At each sampling instant k, the current belief $\mathrm{bel}_p\left(\mathbf{x}(k)\right)$ represents possible values for the state vector – the particles. For each particle $\mathbf{x}_i(k)$, a control signal $\mathbf{u}_i(k)$ is obtained as

$$\mathbf{u}_i(k) = g\left(\bar{\mathbf{x}}_i(k)\right),$$

with $\bar{\mathbf{x}}_i(k)$ computed by (9), leading to

$$\mathrm{bel}_p\left(\mathbf{u}(k)\right) = \{\mathbf{u}_1(k), \mathbf{u}_2(k), \ldots, \mathbf{u}_M(k)\} \tag{10}$$

where $g\left(\bar{\mathbf{x}}_i(k)\right)$ is an appropriate mapping from the state space to the space of inputs.

For each particle, a coordinate change (Lages and Hemerly, 1998) is considered,

$$e = \sqrt{\bar{x}_1^2 + \bar{x}_2^2} \tag{11}$$

$$\psi = \mathrm{atan2}(\bar{x}_2,\, \bar{x}_1) \tag{12}$$

$$\alpha = \bar{x}_3 - \psi, \tag{13}$$

and the input signal $\mathbf{u}_i(k) = [u_{i_1}(k) \;\; u_{i_2}(k)]^{\mathrm{T}}$

$$u_{i_1} = -\gamma_1 e \cos\alpha \tag{14}$$

$$u_{i_2} = -\gamma_2\alpha - \gamma_1 \cos\alpha \frac{\sin\alpha}{\alpha}(\alpha - h\psi), \tag{15}$$

with h, γ_1, $\gamma_2 > 0$, makes (1) asymptotically stable (Lages and Hemerly, 1998). As a consequence, the input belief contains input signals related to point stabilization of the state particles under no state transition uncertainty.

The input belief is obtained by computing (14) and (15) for each state particle. The criterion for choosing an input vector among $\mathrm{bel}_p(\mathbf{u}(k))$ is to select the one with most local support, which means choosing the one whose neighborhood contains the most values also among $\mathrm{bel}_p\left(\mathbf{u}(k)\right)$. The neighborhood of each input vector $\mathbf{u}_i(k)$ was chosen as an ellipsoidal region $\mathcal{S}_i$, centered at $\mathbf{u}_i(k)$, given by

$$\mathcal{S}_i = \left\{ \mathbf{u}(k) : \frac{(u_1 - u_{i_1})^2}{a_1^2} + \frac{(u_2 - u_{i_2})^2}{a_2^2} < 1 \right\},$$

where a_1 and a_2 are the ellipsoid radii which are empirically chosen as functions of the noise in input signals. The ellipsoid form is based on input limits for wheel speeds.

3.3 Extended Kalman Filter

We compare the proposed control system scheme for mobile robots, which uses a particle filter and a could of input signals, with one that uses an extended Kalman filter to estimate its pose, and then the same mapping from states to inputs which was presented in section 3.2.

The Kalman filter depends on the existence of additive Gaussian noise, so it is restricted in the sense of what kind of information can be obtained from sensors: it must, at the very least, have uncertainty terms with a distribution which is close to Gaussian for a viable approximation. In both Kalman filter and particle filters, the output is used to update the state estimate. This step is independent of the state transition prediction and takes place only when $\mathbf{y}(k)$ is available.

Both particle filters and extended Kalman filters consider approximations in order to provide an estimate for the state. Particle filters return an estimation represented by a set of possible values, the particles, which serves as an approximation to a joint probability distribution. The "real" probability distribution can be any. Extended Kalman filters provide an approximation to this PDF that is a multivariate Gaussian. It also approximates the state transition and output equations by their first-order Taylor expansions.

The EKF update takes the Kalman gain and the observation to correct the state estimate. As the belief is summarized by a mean ($\hat{\mathbf{x}}(k)$) and a covariance matrix $\mathbf{Q}(k)$, the update equations are related to those parameters. The required parameters are the Jacobian $\mathbf{F}(k)$ of $f_d(\,\cdot\,)$

$$\mathbf{F}(k) = \left.\frac{\partial f_d(\mathbf{x}(k),\mathbf{u}(k))}{\partial \mathbf{x}(k)}\right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} = \begin{bmatrix} 1 & 0 & -Tu_1(k)\operatorname{sinc}\left(\frac{Tu_2(k)}{2}\right)\sin\left(x_3(k)+\frac{Tu_2(k)}{2}\right) \\ 0 & 1 & Tu_1(k)\operatorname{sinc}\left(\frac{Tu_2(k)}{2}\right)\cos\left(x_3(k)+\frac{Tu_2(k)}{2}\right) \\ 0 & 0 & 1 \end{bmatrix};$$

the Jacobian $\mathbf{H}(k)$ of $h(\,\cdot\,)$,

$$\mathbf{H}(k) = \left.\frac{\partial h(\mathbf{x}(k),\mathbf{u}(k))}{\partial \mathbf{x}(k)}\right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} = \mathbf{I};$$

and the noise covariance matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$ from $f_d(\,\cdot\,)$ and $h(\,\cdot\,)$, in this order. The latter is given by

$$\mathbf{R}(k) = \operatorname{Cov}\left\{\mathbf{v}(k)\mathbf{v}^{\mathrm{T}}(k)\right\},$$

while $\mathbf{Q}(k)$ will demand some approximations, as follows.

Noise is not additive in (4). Moreover, it is also not Gaussian, since it is distorted by a nonlinear mapping (see section 3). By using trigonometric identities in (4), and assuming the noise terms w_t and w_{d_1} are small, we can find an approximation given by

$$\mathbf{x}(k+1) \approx \mathbf{x}(k) + f_d(\mathbf{x}(k),\mathbf{u}(k)) + Tu_1(k)\begin{bmatrix} w_t\cos x_3 - w_{d_1}\sin x_3 \\ w_t\sin x_3 - w_{d_1}\cos x_3 \\ w_{d_1}+w_{d_2} \end{bmatrix}.$$

Let $\Sigma_0(k)$ be the covariance matrix for the noise in x_1 and x_2 for $x_3 = 0$, $u_2 = 0$. By assuming that noises are independent and uncorrelated, it is given by

$$\Sigma_0(k) = \begin{bmatrix} \sigma_1(k) & 0 \\ 0 & \sigma_2(k) \end{bmatrix},$$

with $\sigma_1 = u_1(k)T\sigma_t$ and $\sigma_2 = u_1^2(k)T^2\sigma_{d_1}/2$. The noise terms in x_1, x_2 can be written as a rotation of the 2D noise with angle $x_3 + u_2T/2$, with rotation matrix

$$\Theta(k) = \begin{bmatrix} \cos x_3 & -\sin x_3 \\ \sin x_3 & \cos x_3 \end{bmatrix}.$$

Then the general covariance matrix $\Sigma_2(k)$ over the first two state variables is given by

$$\Sigma_2 = \Theta\Sigma_0\Theta^{\mathrm{T}} = \begin{bmatrix} c^2\sigma_1^2 + s^2\sigma_2^2 & sc\sigma_1^2 - sc\sigma_2^2 \\ sc\sigma_1^2 - sc\sigma_2^2 & s^2\sigma_1^2 + c^2\sigma_2^2 \end{bmatrix}$$

where s and c are $\sin(x_3 + u_2T/2)$ and $\cos(x_3 + u_2T/2)$, respectively. The full covariance matrix $\mathbf{Q}(k)$ is given by

$$\mathbf{Q}(k) = \begin{bmatrix} \Sigma_2 & 0 \\ 0 & T^2u_1^2(k)(\sigma_{d_1}+\sigma_{d_2})^2 \end{bmatrix}.$$

The state estimate update and prediction steps are done according to the usual EKF equations.

4. SIMULATION RESULTS

The simulated robot was modeled as the continuous-time stochastic system (3), being that the state evolution was obtained by 4th order Runge-Kutta with four steps at each sampling instant. As for the particle filter prediction step, the robot was modeled as a discrete-time stochastic (nonlinear) system, described by (4). The values of the noise parameters σ_t and σ_D were set as 0.005 and 0.1745 rad/m, respectively. The observation covariance matrix is $\mathbf{P} = \operatorname{diag}(\sigma_{y_1}, \sigma_{y_2}, \sigma_{y_3})$, with $\sigma_{y_1} = \sigma_{y_2} = 0.1$ m and $\sigma_{y_3} = 1°$. The maximum speed the wheels can achieve is 0.471 m/s. The control sampling period T is 50 ms and the GPS output period is 200 ms. A total of 900 particles were used for the estimation and equally spaced particles inside a square of 1 m^2 centered at $\mathbf{x}(0)$. The controller parameters were $\gamma_1 = 0.5$, $\gamma_2 = 0.5$ and $h = 1.0$. The ellipsoid radii related to u_1 and u_2 are 0.05 and 0.2, in this order. The initial position was $\mathbf{x}(0) = [4\ 0\ \pi]^{\mathrm{T}}$ and the reference $\mathbf{x}_r$ was set to $[1.0\ 3.0\ -\pi/2]^{\mathrm{T}}$.

At each sampling instant, the state belief prediction is carried out, while the belief update takes place when the observation from the GPS is available. Else, there is no further information and the prior belief is taken as the current belief.

The set of input signal particles is obtained from the current belief. This has a much different shape than the state belief, as the mapping from states to control signals is done through a discontinuous coordinate transformation between the system coordinates, which then is subject to a nonlinear function of such transformed coordinates.

The input belief at $k = 0$ is presented in figure 1. As the particles from the initial state belief are structured in a lattice, the resulting input belief keeps part of that structure.

The state belief at $k = 50$ is presented in figure 2. Figure 3 shows its plot in the $X_1 \times X_2$ plane with orientation omitted. The corresponding input belief is shown in figure 4.

An input signal is selected as the particle that maximizes the number of other particles in its neighborhood, as

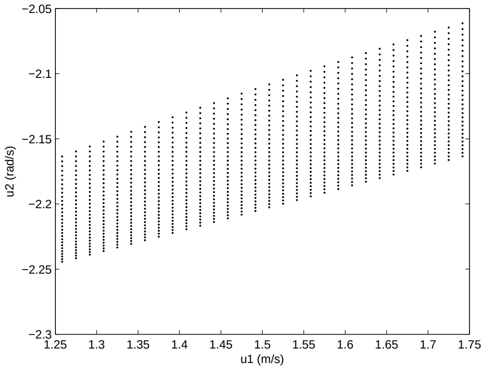

Fig. 1. Input belief at $k = 0$.

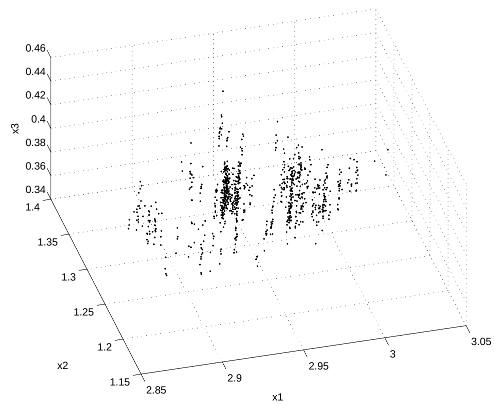

Fig. 2. State belief at $k = 50$.

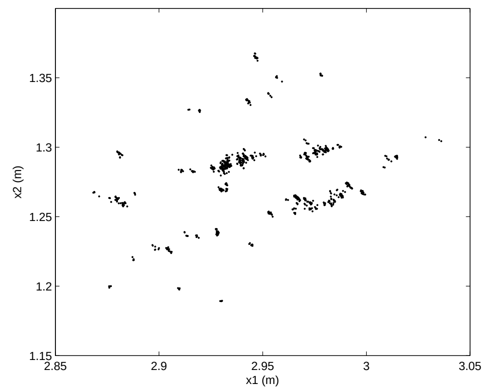

Fig. 3. Projection of the state belief at $k = 50$ in the $X_1 \times X_2$ plane. Orientation is omitted.

explained in Section 3.2. While most of them are related to state particles inside a neighborhood of each other, this is not always true as $g(\mathbf{x}(k))$ is nonlinear and discontinuous. This action can be understood as maximizing the number of vector states that would be driven similarly to the behavior of a deterministic system. We also note that this is an approximation of taking a value related to a region of high probability density, as the particles are an approximation of a continuous joint probability density function as random sparse values. Figure 5 shows the input signals with respect to time.

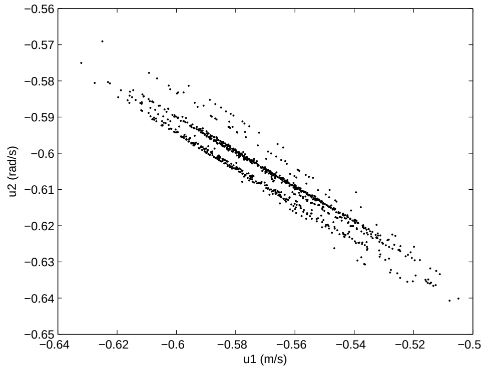

Fig. 4. Input belief at $k = 50$.

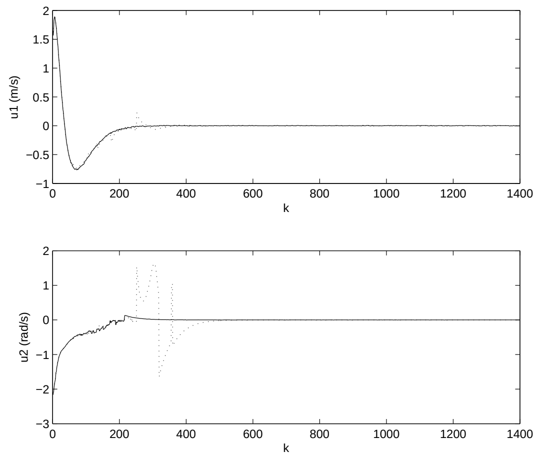

Fig. 5. Input signals with respect to time.Solid line: proposed method. Dotted line: EKF and certainty equivalence.

The trajectory of the robot on the plane is presented in figure 6, along with the simpler method presented in section 3.3. The state-input mapping is such that the robot approaches the reference with small angle error and negative linear speed (see figure 5). This is so due to (11) and (12), which forces $e > 0$ and given that the Lyapunov function has a quadratic term in ϕ. The final position of the robot in Figure 6 is $\mathbf{x}(k) = [1.044\ 3.090\ -1.571]^{\mathrm{T}}$.

The experiment has been repeated in order to verify the stochastic effects in steady state. Particularly, the experiment has been reproduced 50 times and the final state position has been recorded. The mean and standard deviation of the state at the last sampling instant are presented in table 1. Table 2 presents the results when EKF and certainty equivalence were used, together with the same state-input mapping.

5. CONCLUSIONS

A method for controlling a differential-drive mobile robot through output feedback has been proposed. The method can accommodate any kind of uncertainty in state transition and sensing, as long as the respective probability density functions are known.

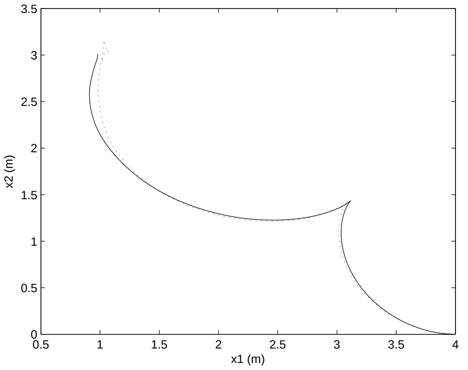

Fig. 6. Robot trajectory of the on the horizontal plane. Solid line: proposed method. Dotted line: EKF and certainty equivalence.

Table 1. Final position mean and standard deviation for $\mathbf{x}(0) = [4\ \ 0\ \ \pi]^{\mathrm{T}}$ and $\mathbf{x}_r = [1\ \ 3\ \ -\pi/2]^{\mathrm{T}}$. Proposed method.

	Mean	Standard deviation
x_1 (m)	0.9973	0.0203
x_2 (m)	3.0027	0.0139
x_3 (rad)	−1.5708	0.0014

Table 2. Final position mean and standard deviation for $\mathbf{x}(0) = [4\ \ 0\ \ \pi]^{\mathrm{T}}$ and $\mathbf{x}_r = [1\ \ 3\ \ -\pi/2]^{\mathrm{T}}$. EKF and certainty equivalence.

	Mean	Standard deviation
x_1 (m)	1.0538	0.2750
x_2 (m)	2.9276	0.2127
x_3 (rad)	−1.4310	0.4977

The system behavior and the information that can be obtained from sensors are subject to uncertainties which are present in mobile robot applications. To these issues, one adds the natural difficulty of controlling non-holonomic systems. In order to overcome these problems an state estimation scheme that uses a particle filter was employed, and the full cloud of state particles was used combined with a non-smooth state feedback law to generate possible input signals which results from the distribution of the state particles.

In order to provide a mean of comparing the method, we have used a more classical feedback method with an extended Kalman filter and a globally stable control law.

The authors are currently working on elaborating and checking different input signal selection policies, and checking how the method behaves for other low-level mobile robotics problems, such as path following. Future work includes testing the method on real robots, considering different types of wheeled mobile robots, and writing parallel code for multicore machines.

REFERENCES

Anderson, B.D.O. and Moore, J.B. (1989). *Optimal Control: Linear Quadratic Methods.* Prentice-Hall Information and System Sciences Series. Prentice-Hall, Englewood Cliffs, NJ, USA.

Astolfi, A. (1994). On the stabilization of nonholonomic systems. In *Proceedings of the 33rd IEEE American Conference on Decision and Control*, 3481–3486. Piscataway, NJ, IEEE Press, Lake Buena Vista, FL.

Baral, C., Kreinovich, V., and Trejo, R. (2000). Computational complexity of planning and approximate planning in the presence of incompleteness. *Artificial Intelligence*, 122(1), 241–267. doi:10.1016/S0004-3702(00)00043-6.

Blanco, J.L., González, J., and Fernández-Madrigal, J.A. (2010). Optimal filtering for non-parametric observation models: Applications to localization and slam. *International Journal of Robotics Research*, 29(14), 1726–1742.

Borenstein, J., Everett, H.R., and Feng, L. (1996). Where am i? - sensors and methods for mobile robot positioning. Technical report, University of Michigan.

Brockett, R.W. (1982). *New Directions in Applied Mathematics.* Springer-Verlag, New York.

Campion, G., Bastin, G., and D'Andréa-Novel, B. (1996). Structural properties and classification of kinematic and dynamical models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1), 47–62.

Kaelbling, L.P., Cassandra, A.R., and Littman, M.L. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1), 99–134. doi:10.1016/S0004-3702(98)00023-X.

Lages, W.F. (1998). *Controle e Estimação de Posição e Orientação de Robôs Móveis.* Tese de doutorado, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP. Orientador: Elder M. Hemerly.

Lages, W.F. and Hemerly, E.M. (1998). Smooth time-invariant control of wheeled mobile robots. In *Proceedings of The XIII International Conference on Systems Science.* Technical University of Wrocław, Wrocław, Poland.

Littman, M.L., Cassandra, A.R., and Kaelbling, L.P. (1995). Efficient dynamic-programming updates in partially observable markov decision processes. Technical Report CS-95-19, Brown University, Providence, RI, USA.

Rekleitis, I.M. (2004). A particle filter tutorial for mobile robot localization. Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Québec, Canada.

Sørdalen, O.J. (1993). *Feedback Control of Nonholonomic Mobile Robots.* Thesis (dr. ing.), The Norwegian Institute of Technology, Trondheim, Norway.

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