# Vehicle Guidance with Control Action Computed by a Rao-Blackwellized Particle Filter

Albert Sans-Muntadas, Edmund Brekke and Kristin Y. Pettersen

***Abstract*— Particle filters can be used in navigation and state estimation problems. They can approximate arbitrary posterior distributions and do not require assumptions of Gaussianity or linearizations. This paper suggests to use the discretized posterior distributions that are produced within the filter beyond the state estimation, to compute control actions based on the information about uncertainty of the states. A case study is presented, and this indicates that the suggested approach can give robustness against uncertainty and reduce the amount of required control action.**

## I. Introduction

The guidance of an autonomous vehicle is often performed based on a sequence of cascaded elements. One is the acquisition of state estimates, then the guidance and control blocks use those estimates to compute control actions that allow the system to achieve the control objectives. The estimates, due to sensor limitations, are often corrupted with noise. Bayesian estimation techniques [1] allow to estimate the states optimally based on the uncertainty of the information, producing states estimates and posterior distributions.

The wide spectra of sequential Monte-Carlo techniques allow to compute any posterior distributions if a sufficiently large amount of particles is given [2], [3]. This makes particle filters versatile for many navigation applications such as terrain or map base location [4].

The posterior density functions of the state vector produced by Bayesian filters are generally used only within the framework of state estimation. It is then implicitly understood that an average or some other statistic from the particle cloud is going to be used further down in the control cascade. However, some authors have pointed out, that controlling a vehicle/robot coherently with awareness of the uncertainty can enhance the robustness and autonomy of the system [5], [6], [7].

There exists an extensive literature on control under stochastic environments [8], [9], [10]. A big part of this literature is concerned about decision processes and dynamic programming. Model Predictive Control can also be implemented for finding trajectories that take into consideration the uncertainty in the state estimate [11], [12], [13], [6].

Contrary to the conventional approach, which computes the control action given the estimate of the state, this paper proposes to use a Rao-Blackwellized particle filter to obtain state estimates; Since the filter also provides a discretized posterior distribution, we can use this information beyond the state estimate problem and use it to compute the corresponding control action (see Fig. 1), similarly to the approach proposed in [12]. This allows us to compute a control action which is in accordance with the state uncertainty. The contribution of the paper lies in the design of a controller formulation that allows the implementation within a Rao-Blackwellized Particle filter. Such a filter reduces the dimension of the sampled state space, when compared to a standard particle filter, making it possible to achieve the same performance with a lower number of particles, thereby decreasing the computational burden.

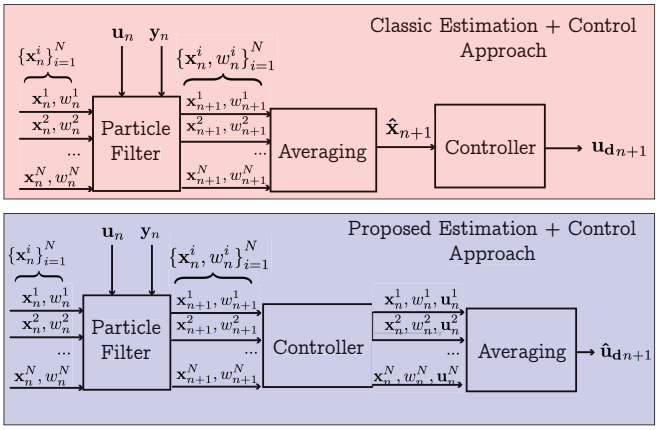


Fig. 1. Block diagram of the architecture of the different control approaches

The paper is organized as follows: Section II formulates the controller and explains the difference between the conventional controller and the proposed one. Section III shows how to calculate the controller based on a Rao-Blackwellized particle filter. Section IV presents a case study to illustrate an implementation and the possible differences between the two control approaches. Section V explains the results and Section VI draws the conclusions.

## II. Controller

A common practice in the implementation of a controller, described by a function $v(\cdot)$, is to use an estimate of the state, typically the expectation value $E[\mathbf{x}] = \int p(\mathbf{x})d\mathbf{x}$, to calculate the desired control input $u = v(E[\mathbf{x}])$. The approach suggested in this paper is to use the expected input $u = E[v(\mathbf{x})]$ as control input instead. For nonlinear functions, these two approaches can have different outcomes

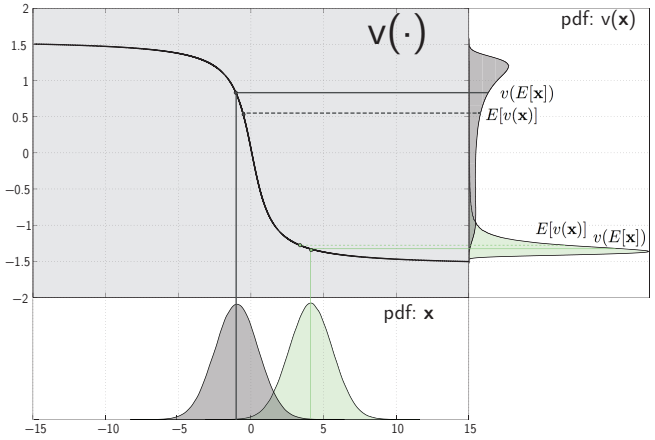


Fig. 2. Example on how computing the expected value of the controller ($E[v(\mathbf{x})]$) or the controller based on the expected value of the states: $v(E[\mathbf{x}])$ lead to differences.

as illustrated in Fig. 2. This is due to how the control function $v(\cdot)$ transforms the probability distribution $p(\mathbf{x})$. Note that $p(\mathbf{x})$ refers to $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ but the conditioning is skipped for notational simplicity.

In order to calculate the value of $E[v(\mathbf{x})]$, the following integral has to be solved:

$$\mathbf{u} = E[v(\mathbf{x})] = \int p(\mathbf{x})v(\mathbf{x})d\mathbf{x} \tag{1}$$

In most of the cases the integral is complex or analytically unsolvable, particularly when the controller, $v(\cdot)$ is nonlinear and the probability distribution is not known in closed form. For this, when a discretization such as a particle filter is used to represent $p(\mathbf{x})$, the expected value can be calculated as [12]:

$$E[v(\mathbf{x})] \approx \hat{E}[v(\mathbf{x})] = \frac{1}{N}\sum_{i=1}^{N} w^{(i)} v(\mathbf{x}^{(i)}) \tag{2}$$

The approach proposed in this paper is designed for a controller that has the following formulation:

$$v(\mathbf{x}) = a_p v_p(\mathbf{x}^p) + a_l V_l \mathbf{x} + a_0 \tag{3}$$

where the state $\mathbf{x}$ can be separated in two parts $\mathbf{x}^p$ and $\mathbf{x}^k$. In particular, $\mathbf{x}^p$ is the part of the state discretized by particles, and $\mathbf{x}^k$ consists of the remaining states. Furthermore, The nonlinear function $v(\cdot)$ has a nonlinear part $v_p(\cdot)$ and a linear part in which $V_l$ is a vector and $a_p, a_l, a_0$ are constants. Then $E[v(\mathbf{x})]$ can be simplified as:

$$\begin{aligned} E[v(\mathbf{x})] =& E[a_p v_p(\mathbf{x}^p) + a_l V_l \mathbf{x} + a_0] \\ =& a_p E[v_p(\mathbf{x}^p)] + a_l V_l E[\mathbf{x}] + a_0 \end{aligned} \tag{4}$$

which by using the approximation described in Eq. (2) can be expressed as:

$$E[V(\mathbf{x})] \approx a_p[\frac{1}{N}\sum_{i=1}^{N} w^{(i)} v_p(\mathbf{x}^{p,(i)})] + a_l V_l \hat{\mathbf{x}} + a_0 \tag{5}$$

Note that $\hat{\mathbf{x}} \triangleq E[\mathbf{x}]$. The linear transformation $V_l\hat{\mathbf{x}}$ does not need a discrete description, while the nonlinear transformation $v(\mathbf{x}^p)$ needs a discrete description. The notation of linear/non-linear parts of the controller separates the states that need a discretized description of the probability density function (pdf). One way to obtain such discretization is by taking advantage of the state estimate produced by a Rao-Blackwellized particle filter (RBPF). The following section explains how to incorporate the controller within the framework of an RBPF.

## III. Implementation of the controller with a Rao-Blackwellized particle filter

Tracking applications often employ Bayesian frameworks to make state estimates. Bayesian estimators combine the knowledge of state-space models and approximations of the posterior density functions to estimate the state vector. A very generic model for Bayesian state estimation [3] can be written as:

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, w_t) \\ \mathbf{y}_t &= h(\mathbf{x}_t, e_t) \end{aligned} \tag{6}$$

At a given time $t$, $\mathbf{x}_t$ is the state variable, $\mathbf{y}_t$ is a measurement, $f, h$ are nonlinear functions and $e_t, w_t$ are measurement noise and process noise, respectively. Then the estimation is done by computing the pdf:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{1:t}) =& \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \\ p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) =& \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t \end{aligned} \tag{7}$$

Depending on whether the integral can be analytically solved, linearized or numerically computed, different filters are used. Among the most common are: Kalman filter (KF), Extended KF (EKF), Unscented KF (UKF), Gaussian sum KF and particle filter (PF).

Particle filters are used when analytical approximations and linearization are difficult to make. Instead, a dynamic stochastic grid of particles is used to represent the posterior density:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{i=0}^{N} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \tag{8}$$

The large grid of particles comes with the drawback of computational expense. The amount of particles needed also grows with the state dimension. Even for moderate dimensions (e.g. as low as 6 ) the problem quickly becomes unfeasible for real time applications [14].

The Rao-Blackwellized PF solves part of this challenge by separating the state vector in two parts: $\mathbf{x}_t^p$ refers to the part of the filter described by particles and $\mathbf{x}_t^k$ represents the one described by KFs. One part enters nonlinearly in the system equations, and thus will be estimated by a particle filter, and one enters linearly in the system equations, and thus will be estimated by a Kalman filter. This reduces the amount of computational burden compared to a basic PF, because

fewer particles are needed due to the smaller dimension of $\mathbf{x}_t^p$. Based on the RBPF formulation proposed in [3], [15] a general expression for the model underlying the RBPF is:

$$\begin{aligned}\mathbf{x}_{t+1}^p &= f^p(\mathbf{x}_t^p) + F^p(\mathbf{x}_t^p)\mathbf{x}_t^k + G^p(\mathbf{x}_t^p)w_t^p \\ \mathbf{x}_{t+1}^k &= f^k(\mathbf{x}_t^p) + F^k(\mathbf{x}_t^p)x_t^k + G^k(\mathbf{x}_t^p)w_t^k \\ \mathbf{y}_t &= h(\mathbf{x}_t^p) + H(\mathbf{x}_t^p)\mathbf{x}_t^k + e_t\end{aligned} \tag{9}$$

For which noises are assumed Gaussian and independent $w_t^p \sim \mathcal{N}(0, Q_t^p)$, $w_t^k \sim \mathcal{N}(0, Q_t^k)$ and $e_t \sim \mathcal{N}(0, R_t)$.

Since the particle filter offers a discretized description of the posterior distribution, it fits nicely to implement the controller proposed in Section II.

The following algorithm, based on the implementation described in [15], illustrates how the computation of the controller (Eq.5) can be implemented within the RBPF:(Note that the model could also include the known control action $u_t$, but this is omitted in the generic expression following the formulation proposed in [15].)

**1.INITIALIZATION**
$N$ Particles are initialized as $\mathbf{x}_{0|-1}^{n,(i)} \sim p_{\mathbf{x}_0^n}(\mathbf{x}_0^n)$ for the nonlinear states and $\{\mathbf{x}_{0|-1}^{k,(i)}, P_{0|-1}^{k,(i)}\} = \{\mathbf{x}_0^k, P_0^k\}$ for the linear states.

**2A. PF MEASUREMENT UPDATE** For the $N$ particles calculate the importance weights

$$q_t^{(i)} = \mathcal{N}\left(h(\mathbf{x}_t^p) + H(\mathbf{x}_t^p)\mathbf{x}_{t|t-1}^k, H(\mathbf{x}_t^p)P_{t|t-1}(H(\mathbf{x}_t^p))^\top + R_t\right) \tag{10}$$

Normalize ($\tilde{q_t}^{(i)} = \frac{q_t^{(i)}}{\sum q_t}$)

$$\hat{\mathbf{x}}_{t|t}^p = \sum \tilde{q}_t^{(i)}\mathbf{x}_t^{p,(i)} \tag{11}$$

**2B. KF MEASUREMENT UPDATE**

$$\begin{aligned}\hat{\mathbf{x}}_{t|t}^k &= \hat{\mathbf{x}}_{t|t-1}^k + K_t\left(\mathbf{y}_t - h(\mathbf{x}_t^p) - H(\mathbf{x}_t^p)\mathbf{x}_{t|t-1}^k\right) \\ P_{t|t} &= P_{t|t-1} - K_t M_t K_t^\top \\ M_t &= H(\mathbf{x}_t^p)P_{t|t-1}(H(\mathbf{x}_t^p))^\top + R_t \\ K_t &= P_{t|t-1}H^\top(\mathbf{x}_t^p)M_t^{-1}\end{aligned} \tag{12}$$

**2C. COMPUTE CONTROL ACTION**

$$u_t = a_p[\sum_{i=1}^{N}\tilde{q}_t^{(i)}v(\mathbf{x}_t^{p,(i)})] + a_l V_l \hat{\mathbf{x}}_{t|t} + a_0 \tag{13}$$

Resample particles if needed.

**3A. PF TIME UPDATE**

$$\begin{aligned}\mathbf{x}_{t+1}^p \sim \mathcal{N}\big(&f^p(\mathbf{x}_t^p) + F(\mathbf{x}_t^p)\mathbf{x}_t^p\ ,\ F^p(\mathbf{x}_t^p)P_{t|t}(F^p(\mathbf{x}_t^p))^\top + \\ &G^p(\mathbf{x}_t^p)Q_t^p(G^p(\mathbf{x}_t^p))^\top\big)\end{aligned} \tag{14}$$

**3B. KF TIME UPDATE**

$$\begin{aligned}\hat{\mathbf{x}}_{t+1|t}^k =& F^k\hat{\mathbf{x}}_{t|t}^k + f^k(\mathbf{x}_t^p) + L_t\left(\mathbf{x}_{t+1|t}^p - f^p(\mathbf{x}_t^p) - F^p(\mathbf{x}_t^p)\hat{\mathbf{x}}_{t|t}^k\right) \\ P_{t+1|t} =& F^k(\mathbf{x}_t^p)P_{t|t}F^k(\mathbf{x}_t^p)^\top + G^k(\mathbf{x}_t^p)Q_t^k(G^k(\mathbf{x}_t^p))^\top - L_t M_t L_t^\top \\ M_t =& F^k(\mathbf{x}_t^p)P_{t|t}F^k(\mathbf{x}_t^p)^\top + G^p(\mathbf{x}_t^p)Q_t^p(G^k(\mathbf{x}_t^p))^\top \\ L_t =& F^k(\mathbf{x}_t^p)P_{t|t}(F^p(\mathbf{x}_t^p))^\top M_t^{-1}\end{aligned} \tag{15}$$

**4. T=T+1 and step 2**

Note that all the states present in the nonlinear part of the controller, $v_p(\mathbf{x}^p)$, must be estimated with a particle filter.

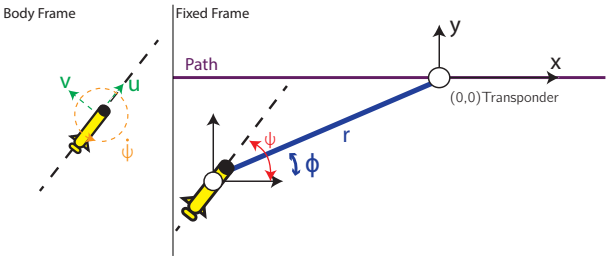


Fig. 3. Representation of the states, the masurements, the coordinates and the path.

Given the case that an RBPF is already being used for state estimation purposes, implementing the controller does not represent a significant increase in the computational burden.

## IV. Illustrative Example and Simulation

The behavior of the proposed control approach is simulated for the case of an underactuated autonomous underwater vehicle (AUV) that moves at a constant velocity. In the proposed scenario the vehicle obtains its navigation information relative to a transponder placed subsea ($p = [0\text{m}, 0\text{m}]$) whose position is assumed to be fixed and known. The AUV measures the distance ($r$) and the bearing angle ($\theta$) from the transponder. The heading direction ($\psi$) is obtained from an internal compass. The mission of the vehicle is to follow a straight line path. The guidance used is line-of-sight (LOS) guidance [16], together with with a proportional controller.

### A. Filter Model

The state vector: $\mathbf{x}_t = [x, y, \psi, \text{u}, \text{v}, \dot{\psi}]^\top$ is used for estimate. The position elements of the state space are estimated by the particle filter part of the Rao-Blackwellized PF, i.e. $\mathbf{x}^p = [x, y]^\top$ The rest of the state is estimated by the Kalman filter: $\mathbf{x}^k = [\psi, \text{u}, \text{v}, \dot{\psi}]^\top$.

The state transition model $F$ used for the filter consists of a constant velocity model and $Q$ represents the process noise of the system. Accelerations are modeled as white noise in this model, and not measured or estimated.

$$\begin{aligned}F &= \begin{bmatrix}1 & 0 & 0 & \text{d}t\cos(\psi) & -\text{d}t\sin(\psi) & 0 \\ 0 & 1 & 0 & \text{d}t\sin(\psi) & \text{d}t\cos(\psi) & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \\ Q &= \text{diag}([10^{-6}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-4}, 10^{-2}]) \\ G &= I_{6x6}\end{aligned} \tag{16}$$

The sensor used for positioning measures range, bearing angle and yaw, that is, $\mathbf{y} = [r, \phi, \psi]^\top$ and $R$ is the covariance matrix of the measurement noise. The observation model that maps true states to measurements is described by the following $H, h$ matrices. The simulated measurements from the USBL sensor that provides range $r$ and bearing, and $\phi$

the magnetometer that provides yaw measurements $\psi$ are corrupted with Gaussian noise.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \; h = \begin{bmatrix} \sqrt{{x_t}^2 + {y_t}^2} \\ \text{atan}\left(\frac{x_t}{y_t}\right) + \pi \\ 0 \end{bmatrix}$$
$$R = \text{diag}([0.2, \tfrac{5\pi}{180}, \tfrac{2\pi}{180}]) \tag{17}$$

### B. Controller

The LOS guidance controller, uses the distance to the path $e$ to calculate the angle at which the AUV should approach the path. The look-ahead distance parameter $\Delta = 3\text{m}$ dictates the ratio of steering compared to the distance to the path. The controller ensures that the vehicle follows the direction prescribed by the guidance law. For simplicity, this example only uses a proportional control action, with $k_p = 10$. For this particular case, the mission is to follow the $x$ axis, therefore $e = y$.

$$\begin{aligned} \psi_d &= \text{atan}\left(\tfrac{-e}{\Delta}\right) \\ u(\mathbf{x}) &= -k_\psi(\psi - \psi_d) \end{aligned} \tag{18}$$

The control action for this example is found by applying to the proposed controller described in Eg. (5) the guidance law and proportional action described in equations (18):

$$u(\mathbf{x}_t) = -k_\psi \left( \sum_{i=1}^{N} -\tilde{q}^{(i)} \, \text{atan}\left(\frac{-y_t^{(i)}}{\Delta}\right)\right) - k_\psi \hat{\psi}_{t|t} \tag{19}$$

### C. Simulation Setup

In order to test and illustrate the difference between the proposed control approach and the classical control approach described in Section II, a batch of 100 simulations is performed for each controller. The simulations use the dynamics of the AUV described in [17].

The first controller uses the conventional approach where the estimate of the position $E[\mathbf{x}]$ is used to calculate the control action: $u = v(E[\mathbf{x}])$. The second, the controller proposed in this paper, finds the expected control action given the position posterior density: $u = E[v(\mathbf{x})]$.

In each simulation the filter uses 1000 particles, and the vehicle is set up to start from the position: $P_0(x, y, \psi) = [-100\text{m}, -5\text{m}, 0\text{rad}]$ and velocity $u, v, \dot{\psi} : [1\text{m/s}, 0\text{m/s}, 0\text{rad/s}]$. The simulations first run the vehicle in a straight line path at $y = -5\text{m}$ to allow the filter to converge; this is done to avoid that the initialization choice of the filter affects the behavior of the controller. Once it converges, at approximately $x = -70\text{m}$, the vehicle starts the mission of following the straight line $y = 0$. The results presented in the next section illustrate how the vehicle behaves, once it has switched to follow the $y = 0$ line.

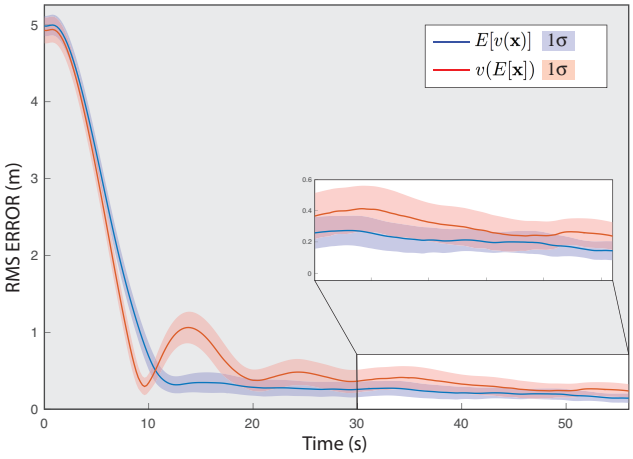


Fig. 5. RMS error. Each color shows the average of 100 simulations and the shaded area represents the standard deviation between simulations.

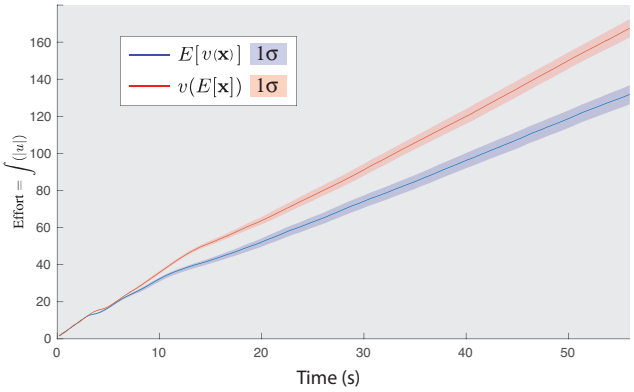


Fig. 6. Metric of the controller's effort.

## V. Simulation Results and Discussion

An example of the performance of The Rao-Blackwellized PF and the guidance is shown in Fig. 4. The vehicle trajectory, represented with a black line, shows that the vehicle is able to converge to and follow the path. The blue line shows the estimate of the position by the RBPF. The figure also shows seven different clouds of particles, obtained at different times during the simulation. One can see that the posterior distribution is quite wide, and there is significant uncertainty regarding whether the vehicle is above or below the path.

The results of the batch of simulations are shown in Fig. 5 and 6: red for the conventional controller, and blue for the proposed controller. In both cases the parameters are tuned equally for the filters and the controllers. Two apparently similar controllers show a slight different behavior, still both of them achieve the control objective of following the straight line path at $y = 0$.

The proposed controller (see Fig. 5, shaded in blue) has a slightly slower steering towards the path. An explanation of this slower convergence rate could have been credited

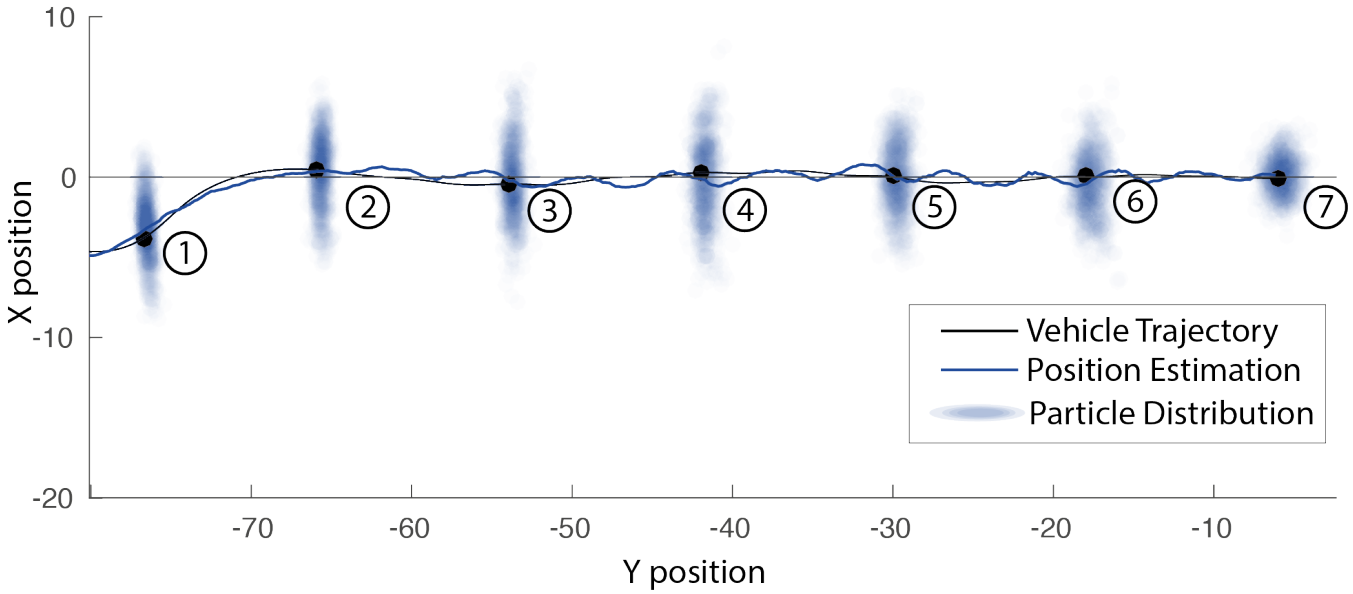


Fig. 4. Result of a simulation the cloud of particles represents the position posterior distribution

by Jensen's inequality that says that for a concave function $E[v(\mathbf{x})] \leq v(E[\mathbf{x}])$. The absolute value of our function $v(\cdot)$ is concave for the intervals $(-\infty, 0)$ and $(0, \infty)$, which means that at least during the approach (when all the pdf lies inside the intervals) the control action is smaller. It can also be interpreted by looking at Fig. 4 at the first point. Here some of the particle distribution lies on the upper side of the path, and those particles contribute to making the controller action smaller than for the conventional controller which uses the state estimate to calculate the control action. On the other side, the position uncertainty gives the proposed controller a more cautious response, which results in less oscillations from side to side. This makes the steady state RMS error smaller than that of the conventional controller. Note also that as the vehicle gets closer to the origin, the posterior distribution becomes narrower (see Fig. 4). This partially explains the reduction of the average RMS error for the conventional controller (red line Fig. 5), since the narrower the posterior distribution the more similar the controllers become.

One of the benefits of using the proposed control approach for this particular case is that it reduces the amount of control action used to steer the vehicle, which is measured as $\int |u| dt$. Fig. 6 shows the the amount of "energy" used by the controller grows slower for the suggested controller.

## VI. Conclusions

This paper has proposed a controller that uses the framework of a Rao-Blackwellized particle filter to compute the expected control action given the uncertainty of the vehicle location. The simulations for the tested scenario show that this control approach may result in a more robust controller and reducing the amount of action required to steer the vehicle. This paper has demonstrated that for the given case, accounting for the uncertainty in the states can improve the performance of a controller.

## References


[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[3] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[4] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *AAAI*, 2002, pp. 593–598.

[5] D. Salmond and N. Gordon, "Particles and mixtures for tracking and guidance," in *Sequential Monte Carlo Methods in Practice*. Springer, 2001, pp. 517–532.

[6] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[7] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, p. 93, 2000.

[8] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[9] H. J. Kushner, "Stochastic stability," in *Stability of stochastic dynamical systems*. Springer, 1972.

[10] H. Pham *et al.*, "On some recent aspects of stochastic control and their applications," *Probability Surveys*, vol. 2, no. 506-549, pp. 1–2, 2005.

[11] N. Dadkhah and B. Mettler, "Survey of motion planning literature in the presence of uncertainty: considerations for UAV guidance," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 233–246, 2012.

[12] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[13] D. S. Bayard and A. Schumitzky, "Implicit dual control based on particle filtering and forward dynamic programming," *International journal of adaptive control and signal processing*, vol. 24, no. 3, p. 155, 2010.

[14] F. Daum, "Nonlinear filters: Beyond the Kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[15] G. Hendeby, R. Karlsson, and F. Gustafsson, "A new formulation of the Rao-Blackwellized particle filter," in *IEEE/SP Statistical Signal Processing*. IEEE, 2007, pp. 84–88.

[16] T. I. Fossen, M. Breivik, and R. Skjetne, "Line-of-sight path following of underactuated marine craft," *IFAC Proceedings*, vol. 36, no. 21, pp. 211–216, 2003.

[17] T. T. J. Prestero, "Verification of a six-degree of freedom simulation model for the remus autonomous underwater vehicle." Masters thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2001.