

- [5] E. Rimon and D. E. Koditschek, "The construction of analytic diffeomorphisms for exact robot navigation on star worlds," *Trans. Amer. Math. Soc.*, vol. 327, no. 1, pp. 71–115, 1991.
- [6] H. Tanner, S. Loizou, and K. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Feb. 2003.
- [7] H. Tanner and A. Kumar, "Towards decentralization of multi-robot navigation functions," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 4132–4137.
- [8] G. A. D. Lopes and D. E. Koditschek, "Level sets and stable manifold approximations for perceptually driven non-holonomically constrained navigation," *Adv. Robot.*, vol. 19, no. 10, pp. 1081–1095, 2005.
- [9] D. C. Conner, A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, 2003, vol. 4, pp. 3546–3551.
- [10] L. Yang and S. M. LaValle, "The sampling-based neighborhood graph: A framework for planning and executing feedback motion strategies," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 419–432, Jun. 2004.
- [11] N. J. Cowan and D. E. Chang, "Geometric visual servoing," *IEEE Trans. Robot. Autom.*, vol. 21, no. 6, pp. 1128–1138, Dec. 2005.
- [12] D. G. C. Handron, "The Morse complex for a Morse function on a manifold with corners," arXiv number: math.GT/0406486, 2006 [Online]. Available: <http://www.arxiv.org>
- [13] S. A. Vakhrameev, "Morse lemmas for smooth functions on manifolds with corners," *J. Math. Sci.*, vol. 4, pp. 2428–2445, 2000.
- [14] M. W. Hirsch, *Differential Topology*. New York: Springer-Verlag, 1976.
- [15] D. E. Koditschek, "The control of natural motion in mechanical systems," *ASME J. Dyn. Syst. Meas. Control*, vol. 113, no. 4, pp. 547–551, 1991.
- [16] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 521–533, Aug. 2002.
- [17] M. Goresky and R. MacPherson, *Stratified Morse Theory, Ser. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Berlin, Germany: Springer-Verlag, 1988, vol. 14.
- [18] E. A. Jonckheere, *Algebraic and Differential Topology of Robust Stability*. New York: Oxford Univ. Press, 1997.

How to Tell a Bad Filter Through Monte Carlo Simulations

Lingji Chen, *Member, IEEE*, Chihoon Lee, and
Raman K. Mehra, *Fellow, IEEE*

Abstract—In this note, we propose one particular method to address the issue of how to numerically evaluate nonlinear filtering algorithms and/or their software implementations, through Monte Carlo simulations. We introduce a quantitative performance indicator whose computation can be automated and does not depend on any specific definition of point estimate. The method is based on conditional probability integral transform and maximum deviation of an empirical cumulative distribution function from a uniform distribution. The usefulness of such an indicator is illustrated through an example.

Index Terms—Algorithm, conditional cumulative density function, density evaluation, implementation, Kolmogorov–Smirnov goodness-of-fit test, Monte Carlo simulations, nonlinear filtering, performance, probability integral transform.

I. INTRODUCTION

In this note, we propose one particular method to address the issue of how to numerically evaluate nonlinear filtering algorithms and/or their software implementations, through Monte Carlo simulations. More specifically, our objective is to quantitatively define an indicator of the filtering performance such that its computation can be automated and its value can suggest incorrect algorithm and/or implementation. Equally important is the requirement that this indicator is not dependent upon any particular definition of point estimate; it should evaluate the filtering probability density *as a whole*.

Generally speaking, the objective of filtering is to recursively obtain good estimates of the unknown "signal X " given its noisy "measurement Y ." Mathematically, the problem is often formulated as first obtaining (approximating) the posterior probability distribution $P(X|Y = y)$ and then defining a "point estimate \hat{X} " of the signal X based on the above distribution. If we have followed both steps, then we can examine the "point estimation error $e \triangleq \hat{X} - X$ " and check whether it is too large.

However, it is sometimes beneficial to treat the above two steps separately. The posterior $P(X|Y = y)$ represents all the information available about X after receiving all measurement up to $Y = y$. This information may be utilized in more than one way in terms of revealing what the true value of X is. In other words, when the point estimation error e is large, it may be the case that the definition of the point estimate is not a good way to utilize the information contained in the posterior. Thus, our focus in this note is on the whole posterior distribution, and not on any statistic computed from the same.

To evaluate a filtering algorithm through Monte Carlo simulations, we start with simulating a signal trajectory X and a measurement trajectory Y , by generating the necessary noise sequences using algorithm

Manuscript received April 26, 2006; revised January 16, 2007. Recommended by Associate Editor I.-J. Wang. This work was supported in part by the U.S. Army STTR Contract #W911NF-04-C-0108, under the direction of Dr. M.-H. (Harry) Chang.

L. Chen and R. K. Mehra are with the Scientific Systems Company, Inc., Woburn, MA 01801 USA (e-mail: chen@ssci.com; rkm@ssci.com).

C. Lee is with the Department of Statistics and Operations Research, University of North Carolina, Chapel Hill, NC 27599 USA (e-mail: chlee@email.unc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2007.900835

mically produced random numbers. After this, we perform filtering on this pair of trajectories and obtain a posterior P conditioned on Y . If we repeat the process, we obtain a different signal trajectory X' with a different measurement trajectory Y' , and, consequently, a different posterior P' , which is conditioned on Y' . How should we aggregate the results, and what should we look for, in order to reveal inconsistency between simulated trajectories and the supposed posteriors, if it so happens that the filtering algorithm is poor or its implementation is wrong?

Before we continue, it is worth pointing out that we are *not* interested in comparing a filtering result with “the ground truth” in the following fashion. First, turn a signal trajectory X into some Dirac’s δ -like “ground truth distribution,” and then measure how much our filtered distribution deviates from that, by using for example Kullback–Leibler divergence. Such a divergence measure may be very useful in comparing different filters, but for one filter we may not be able to tell whether it is suspiciously too large. Our interest is in answering the question: After we have conducted Monte Carlo simulations and collected many triple sets $\{X, Y, P(\cdot|Y)\}$, and there has been a mistake/bug in the algorithm/software, should we not be able to tell from the data?

Our answer comes from an intuitive observation that X can be thought of as a “sample” from $P(\cdot|Y)$, and can be further transformed, through probability integral transform, into a sample Z from the uniform distribution. Then we can aggregate all the Z s and see whether there is any inconsistency.

We will present such a result in the framework of conditional distributions. To the best of the authors’ knowledge, the idea is a new one in the context of evaluating filtering algorithms. Literature search reveals that there has been intensive research on using probability integral transform to evaluate density forecasts in the econometrics community; see [1] and the references therein.

This note is organized as follows. First, a qualitative description is given in Section II, for the benefits of readers who may not be very familiar with measure-theoretic formulation of probability theory. In Section III, the underlying mathematical concept is introduced through two theorems: Theorem 1 is the well known result on probability integral transform. Theorem 2 is new (as far as the authors know) in its formulation for repeated experiments; literature in econometrics gives results in terms of independent and identically distributed (i.i.d.) sequences as time evolves. Section IV applies Theorem 2 and its corollary to the evaluation of filtering density, and defines a “badness indicator” as the maximum deviation from the uniform distribution among a set of scalar valued random variables “projected down” from the multi-dimensional signal X . The usefulness of this indicator is illustrated in Section V with a simple numerical example, where we can envision the optimal posterior density. It is shown that the indicator indeed prominently reflects the difficulty encountered by the extended Kalman filter in giving a unimodal approximation of a bimodal posterior. Conclusions are drawn in Section VI.

II. QUALITATIVE DESCRIPTION OF THE RESULT

Before we present the precise mathematical result in the next section, we describe here the intuition behind it and how it is used to construct a “badness indicator” for filtering algorithms.

First, in the filtering problem we consider, we will fix an end time T in obtaining the posterior distribution of $[x_0; x_1; \dots; x_T]$ given measurements $[y_1; \dots; y_T]$. Thus, the essence of our approach can be illustrated by talking about two random variables X and Y rather than two sequences coming from state evolution and measurement of a dynamical system, and we will do that to make the exposition simpler.

If we are given a set of samples $\{X^{(i)}\}$, $i = 1, \dots, N$, and a claim that they are all drawn from a single distribution $P(\cdot)$, then there are

tests that we can conduct to see how likely the claim is true. Intuitively, suppose $P(\cdot)$ is Gaussian, we can plot the histogram of $\{X^{(i)}\}$ and see whether it, indeed, resembles a bell shaped curve. Later, we will introduce Kolmogorov–Smirnov goodness-of-fit test for this purpose.

When we conduct simulations of a filtering algorithm N times, what we get is a collection $\{X^{(i)}, Y^{(i)}, P(\cdot|Y^{(i)})\}$, $i = 1, \dots, N$, where the triplet represents signal, measurement and the corresponding posterior distribution respectively. Suppose $P(\cdot|Y^{(i)})$ is, indeed, the correct answer. Then $X^{(i)}$ can be thought of as a sample drawn from the distribution $P(\cdot|Y^{(i)})$, or $P(\cdot|i)$ for notational convenience. Since both are changing from simulation to simulation, we need to somehow transform the data such that we can conduct goodness-of-fit test as described above. Probability integral transform does the job: It is well known that if $F(\cdot)$ is the continuous cumulative distribution function of a random variable X , then the random variable $F(X)$ is distributed uniformly on the interval $(0, 1)$. Thus, from $P(\cdot|i)$, we can get the cumulative distribution $F^{(i)}(\cdot)$, and transform $X^{(i)}$ into $Z^{(i)} \triangleq F^{(i)}(X^{(i)})$. Now we can test whether $\{Z^{(i)}\}$ is a set of samples drawn from the uniform distribution over the interval $(0, 1)$. If the discrepancy is large, there is reason to suspect that $P(\cdot|i)$ is not really the correct answer.

This note presents a *precise* mathematical statement of the above result and its proof.

The probability integral transform is applicable only to scalars. If X is a vector, then we can choose several scalar valued functions $g_1(X), g_2(X), \dots$, and $g_L(X)$ and apply the probability integral transform to generate L sets of samples, each of which is supposedly drawn from the uniform distribution. We can then take the largest deviation among them as a “badness indicator,” which we will define later in the note.

In summary, we can add two routines to our Monte Carlo simulations in order to set up a “badness indicator” as a warning signal for quality control. One routine takes $X^{(i)}$ and a chosen scalar valued function $g(\cdot)$, and gives a transformed value $Z_g^{(i)}$ based on the posterior distribution $P(\cdot|i)$. The other routine takes a set of (scalar-valued) samples and gives its deviation from a uniform distribution, as defined later in the note. Whenever the largest deviation exceeds a threshold, a warning can be issued so that the algorithm/software can be re-examined.

Even though knowledge of the underlying model and distributions is required to perform the necessary Monte Carlo simulations, the calculation involved in the above “badness indicator” can be automated, making it useful for cases in which complex models and nontrivial distributions would prevent a more direct analysis of the filtering results for many practical applications.

III. PROBABILITY INTEGRAL TRANSFORMS

Let $\{\Omega, \mathcal{F}, P\}$ be a probability space and $X : \Omega \rightarrow \mathbb{R}$ a random variable. The cumulative distribution function (CDF) of X is defined for $x \in \mathbb{R}$ as

$$F(x) \triangleq P(\{\omega \in \Omega : X(\omega) \leq x\}).$$

The following probability integral transform theorem is well known. The proof is given in Appendix A to help readers draw parallels between this and the proof of Theorem 2.

Theorem 1: If the CDF $F(\cdot)$ of the random variable X is continuous, then the random variable $Z \triangleq F(X)$ is uniformly distributed on the interval $(0, 1)$.

Proof: See Appendix A.

Introduce another random variable $Y : \Omega \rightarrow \mathbb{R}$. Our objective is to define the conditional CDF, but it will be a bit involved. For $x \in \mathbb{R}$, define

$$A(x) \triangleq \{\omega \in \Omega : X(\omega) \leq x\}.$$

Let $I_{A(x)}$ be the indicator function of the set $A(x)$, i.e., $I_{A(x)}(\omega) = 1$ if $\omega \in A(x)$, and 0 otherwise. The conditional expectation of the random variable $I_{A(x)}$ with respect to the event $Y(\omega) = y$ can be defined [2] as a Borel function $m_x(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\int_{\{\omega: Y(\omega) \in B\}} I_{A(x)} P(d\omega) = \int_B m_x(y) P_Y(dy) \quad (1)$$

for any $B \in \mathcal{B}(\mathbb{R})$, the Borel σ -algebra on \mathbb{R} , where $P_Y(\cdot)$ is the measure induced by Y . We will rewrite $m_x(y)$ as $M(x, y)$.

We make the following simplifying assumption regarding $M(\cdot, \cdot)$.

Assumption 1: For every $y \in \mathbb{R}$, $M(x, y)$ is a continuous CDF in x .

Remark: Conditional expectation is uniquely defined only up to sets of measure zero; hence, its many properties hold only almost surely. In general, $M(\cdot, y)$ may not be a CDF for every y ; properties of CDF may fail to hold on a set of measure zero. Thus, Assumption 1 can be more precisely stated as that there exists a Borel measurable function $M(\cdot, \cdot) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ such that for every y , $M(x, y) = m_x(y)$ is a continuous function in x and (1) holds for every x . The assumption is clearly satisfied if X and Y admit a joint density.

With this, we have effectively defined the conditional CDF of X given $Y = y$ to be

$$P(X \leq x | Y = y) \triangleq M(x, y).$$

Theorem 2: Let $M(x, y)$ be defined as above with Assumption 1. Then $Z(\omega) \triangleq M(X(\omega), Y(\omega))$ is a random variable uniformly distributed on $(0, 1)$.

Proof: See Appendix B and C.

Corollary 1: The above theorem extends to the case when Y is a random vector, i.e., $Y : \Omega \rightarrow \mathbb{R}^m$.

IV. FILTERING DENSITY EVALUATION

Let us now apply the above results to a nonlinear filter. Let $\{X_t\}, t \in \mathbb{N}$ be the signal process defined on the probability space (Ω, \mathcal{F}, P) and taking values in \mathbb{R}^n , and $\{Y_t\}, t \in \mathbb{N}$ be the measurement process taking values in \mathbb{R}^m . Define the shorthand notation $X_{0:t} = [X_0; X_1; \dots; X_t]$ where semicolon denotes concatenation of column vectors. Similarly, define $Y_{1:t}$. The objective of filtering is to obtain the conditional distribution of $X_{0:t}$ given the σ -algebra of $Y_{1:t}$. In this note, we fix $T \in \mathbb{N}$ and are interested only in the marginal conditional distribution of X_T , which is denoted by $\pi_T(\cdot)$. Since we need scalar “projections” of X_T , let $g(\cdot)$ be a scalar valued continuous function² of X_T . We assume that the conditional CDF of the scalar random variable $g(X_T)$, as defined in the previous section, is continuous (see Assumption 1). We denote it by $M_g(x, Y_{1:T})$, and note that

$$M_g(x, Y_{1:T}) = \int_{\{\xi \in \mathbb{R}^n: g(\xi) \leq x\}} \pi_T(d\xi).$$

Thus, according to Theorem 2 and its corollary, the random variable

$$Z_T(\omega) = M_g(g(X_T(\omega)), Y_{1:T}(\omega)) \quad (2)$$

is uniformly distributed over $(0, 1)$.

The implication of the above is as follows. We are interested in evaluating a particular nonlinear filtering algorithm, which gives $\hat{\pi}_T$ as an approximation to the optimal filter π_T . For one Monte Carlo simulation that we conduct of both the signal process and the observation process, we can compute the approximate filter and subsequently the

¹One can start with this as a *definition*, or one can show this as a *consequence* of an equivalent definition; see [2, p.220].

²Note that $g(\cdot)$ can be more general, as long as the conditional CDF of $g(X_T)$ is continuous.

transformed value $Z_T^{(1)} \triangleq Z_T(\omega_1)$ from (2). Repeat the simulation N times, and we obtain the set

$$S_g = \left\{ Z_T^{(i)} \right\}, \quad i = 1, 2, \dots, N \quad (3)$$

for a particular choice of $g(\cdot)$. If the filtering algorithm approximates the optimal filter well, then the above set should be close to a sample from the uniform distribution. A large deviation indicates a bad filtering algorithm or an incorrect software implementation.

Motivated by Kolmogorov–Smirnov goodness-of-fit test [3], we quantitatively define the deviation through the use of an *empirical cumulative distribution function* (ECDF). More specifically, let the unit step function be defined as

$$u(x) \triangleq \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$$

The ECDF $F_S(x)$ of a set $S \triangleq \{z_i\}, i = 1, 2, \dots, N$ is defined as

$$F_S(x) \triangleq \frac{1}{N} \sum_{i=1}^N u(x - z_i).$$

This is a step function that increases by $1/N$ at every point z_i . Its deviation from a postulated CDF $F(x)$ is defined as the maximum of their absolute pointwise differences. Note that the CDF of a uniform distribution is a straight line $F(x) = x$ over $(0, 1)$. Hence, we define the deviation

$$D_g \triangleq \max_{x \in [0, 1]} |F_{S_g}(x) - x|$$

where the set S_g is defined by (2) and (3).

We want to “project down” X_T through more than one $g(\cdot)$. Let

$$G \triangleq \{g_1(\cdot), \dots, g_L(\cdot)\} \quad (4)$$

be a set of predefined scalar valued continuous functions.³ We define a *Badness Indicator* for a nonlinear filtering algorithm as

$$D \triangleq \max_{g \in G} D_g. \quad (5)$$

If this value is large, then there is reason to suspect that the nonlinear filtering algorithm or its implementation is problematic. We point out two features that this indicator enjoys.

- 1) Its calculation can be automated.
- 2) It does not depend on a definition of point estimate of X_T ; it checks whether the filtered density as a whole is consistent with simulated trajectories.

Remark: The distribution of the values of D as defined in (5) is hard to determine. Hence, unlike in Kolmogorov–Smirnov goodness-of-fit test, we will not go into hypothesis testing here. Instead, we will use the badness indicator merely as a “warning signal” in testing and evaluating a particular filtering algorithm and/or its software implementation. This will be illustrated in the next section.

V. EXAMPLE

We will use an example to demonstrate the calculation of the badness indicator and its potential usefulness. The example is chosen such that:

- using our “intelligence,” we can tell the right answer and the wrong answer by mere inspection, while on the other hand;
- using the “dumb” badness indicator, we can receive a suitable warning against the wrong answer.

³Here we are “scanning” the multidimensional X_T from one-dimensional “angles.” Alternatively, we can utilize transformations such as one reported in [4] to examine the transformed X_T in a hypercube.

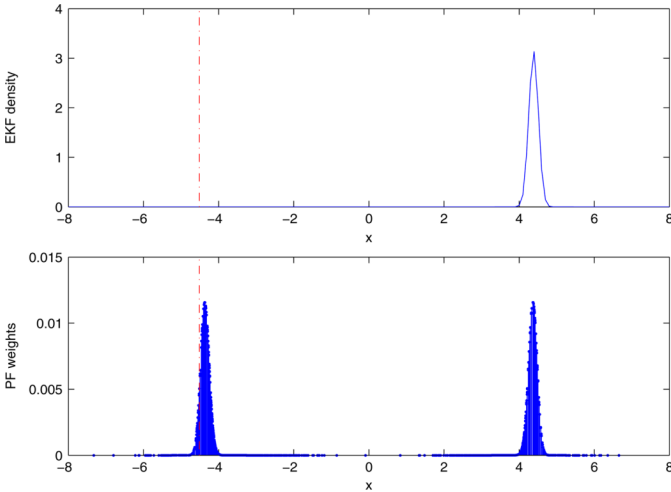


Fig. 1. One particular run of the simulation. Dashed line indicates the true position on the x axis. Top plot shows the density of the EKF Gaussian, while bottom plot shows the weights of the 1000 particles.

If the badness indicator can, indeed, achieve the latter, then it can be useful for complex systems for which we cannot easily tell what the right answer is.

We consider the following discrete-time system:

$$x_k = x_{k-1} + w_{k-1} \quad (6)$$

$$y_k = x_k^2 + v_k. \quad (7)$$

Here, x_k and y_k are scalars representing signal and measurement, respectively, and v_k and w_k are i.i.d. Gaussian noises. Fix $T = 10$, and we are interested in the posterior distribution $P(x_T | y_{1:T})$. We consider two approximations, one based on extended Kalman filter (EKF) and denoted by π_1 , the other based on particle filter (PF) and denoted by π_2 . More specifically

$$\pi_1 \sim \mathcal{N}(\mu, \sigma^2), \quad \pi_2(dx) = \sum_{i=1}^L w_i \delta_{\xi_i}(dx)$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2 , the set $\{\xi_i, w_i\}, i = 1, 2, \dots, L$ is a set of samples (particles) with weights, and $\delta_x(\cdot)$ is the Dirac's delta function centered at x .

If we inspect the measurement (7) and Fig. 1, where one particular run of the Monte Carlo simulations is shown, it can be seen that the nonlinearity is such that the sign of x_k cannot be resolved. Intuitively we know that the posterior distribution should have two modes, as is the case with the PF shown in the bottom plot in Fig. 1. The EKF, on the other hand, gives a unimodal Gaussian approximation, which is shown in the top plot in Fig. 1. In both plots the “true” value of the state x_T in this particular run is indicated by a vertical (red) dash-dotted line.

We conclude that the result from PF is most likely correct, while the result from EKF is not correct or at least problematic. Note that we arrived at our conclusion by employing an understanding of the nonlinearity in (7). Note also that we did *not* arrive at our conclusion by calculating the mean squared error of some point estimate obtained from the posterior distribution. In fact, if we are forced to pick a point estimate from PF, its performance would not be much better than that of EKF: Picking the particle mean $\sum_{i=1}^L w_i \xi_i$ does not yield much information since it is almost always zero, while picking one of the two modes will result in missing x_T when it is at the other mode, which happens quite frequently.

The badness indicator proposed in the previous section does not depend upon “human understanding” of the equations, once the Monte

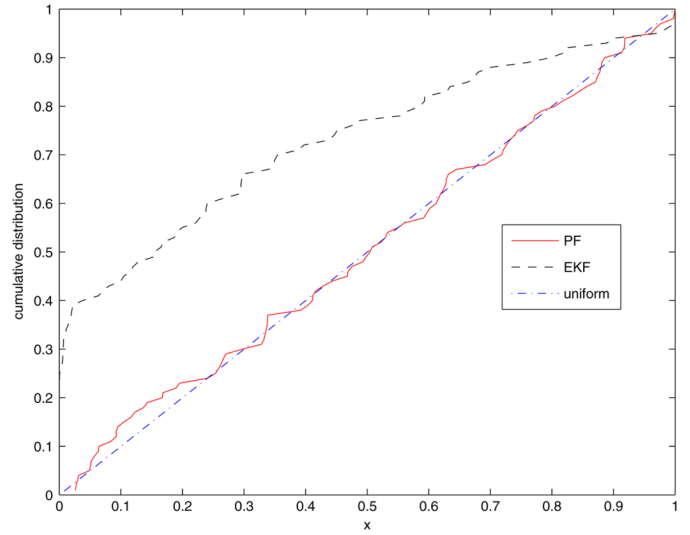


Fig. 2. ECDF for PF, EKF, and the uniform distribution. The deviation of the EKF curve from the diagonal line is large, indicating badness of the filter.

Carlo simulation is constructed. It can be computed in an automated fashion, and is not based on any definition of point estimate. The computation procedure is as follows.

- 1) For each simulation, pick an initial condition x_0 from its initial distribution which is assumed to be $\mathcal{N}(1, 1)$. Draw noise sequences w_k and v_k from their distributions, respectively, in this case, both assumed to be $\mathcal{N}(0, 1)$.
- 2) Simulate the state trajectory and the measurement trajectory.
- 3) Perform EKF on this pair of trajectories to obtain $\mathcal{N}(\mu, \sigma^2)$.
- 4) Perform PF on the same pair of trajectories to obtain the set $\{\xi_j, w_j\}, j = 1, 2, \dots, L$, where L is chosen to be 1000. The algorithm is described as the “Bootstrap Filter” in [5, p.11]. Note that resampling is done in each step, but only after inference is made [6]. In other words, the weight values shown in Fig. 1 will be resampled and become uniform when filtering proceeds to the $T + 1$ step.
- 5) Calculate the CDF-transformed point for the EKF

$$z_{\text{EKF}} = \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{x_T - \mu}{\sqrt{2}\sigma} \right) \right)$$

where $\operatorname{erf}(\cdot)$ is defined as

$$\operatorname{erf}(z) \triangleq \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$$

and is available in Matlab and other numerical software.

- 6) Calculate the CDF-transformed point for the PF

$$z_{\text{PF}} = \sum_{\ell \in \{\ell: \xi_\ell \leq x_T\}} w_\ell.$$

This is used as an *approximation* to the value that should be obtained from a smoothed (regularized) version of PF (see Assumption 1). Since we are using $L = 1000$ particles, this approximation is already good enough for the purpose of obtaining the badness indicator.

- 7) Repeat the simulations for $N = 1000$ times to collect $\{z_{\text{EKF}}^{(j)}\}$ and $\{z_{\text{PF}}^{(j)}\}, j = 1, 2, \dots, N$.
- 8) Obtain ECDF for each set, respectively, and calculate the maximum deviation from that of a uniform distribution.

The two ECDFs for EKF and PF are shown in Fig. 2, as well as the CDF of a uniform distribution. The figure indicates that there is large in-

consistency between the EKF results and simulated trajectories, while PF does not suffer from this problem. This agrees with our observation of simulation results as illustrated in Fig. 1

The signal x in this example is a scalar. For the multidimensional case, we can choose linear projections of x as the scalar functions in the definition of the set G in (4), i.e., $g_\ell(x) \triangleq \eta_\ell^T x$ where η_ℓ is some vector of unit length. For EKF, the projected distribution is Gaussian. For PF, the projected distribution can be expressed with the projected particles. Hence, both cases are easy to compute.

It is worth pointing out that the above simulation result does not mean that PF is correct in approximating the optimal posterior density. Being consistent with simulated trajectories is only *necessary* for a filter to be a good one. It is *not* sufficient. To see this, consider the case when the initial distribution of x_0 is assumed $\mathcal{N}(0, 1)$ instead of $\mathcal{N}(1, 1)$. To simplify our discussion, let $T = 1$. In EKF, after the prediction step using (6), the predicted distribution of x_1 is $\mathcal{N}(0, 2)$ since the distribution of w_0 is assumed to be $\mathcal{N}(0, 1)$. The linearized equation for (7) evaluated at $x_1 = 0$ is $y_1 = (2 \cdot 0)x_1 + v_1$, which does not provide information on x_1 . Hence, after the correction step, the posterior distribution of x_1 is still $\mathcal{N}(0, 2)$. It is always consistent with x_1 ; the badness indicator is very small. However, EKF is far from a good approximation to the optimal filter, since it does not incorporate any information contained in the measurement.

VI. CONCLUSION

In this note, we have proposed one particular method to address the issue of how to numerically evaluate nonlinear filtering algorithms and/or their software implementations, through Monte Carlo simulations. We have quantitatively defined a performance indicator that can be computed with automated routines and does not depend on any particular definition of point estimate. The method is based on conditional probability integral transform and maximum deviation of an empirical cumulative distribution function from a uniform distribution. The usefulness of such an indicator is illustrated through an example.

This indicator can be used as a “quality control routine” for any new filtering algorithm/implementation. A large value of the indicator raises a warning sign so that further inspection can be made. It also complements methods based on point estimations: If filtering density is consistent, then a large estimation error may have arisen from inappropriate definition of the point estimator.

In this note, we have fixed the end time $t = T$ and examined the associated posterior through repeated simulations. It is also perceivable that a similar method can be developed for *one simulation*, to examine the realized trajectory versus the calculated posterior as time t progresses. More research is needed if such results were to be rigorously established.

APPENDIX

A. Proof of Theorem 1

Proof: This proof is similar to the one given in [7]. Let $u \in (0, 1)$. Define

$$\tau(u) \triangleq \sup\{x \in \mathbb{R} : F(x) \leq u\}.$$

If $F(\cdot)$ is continuous, then

$$F(\tau(u)) = u.$$

Combining with the fact that $F(\cdot)$ is nondecreasing, we conclude that

$$\{\omega : F(X(\omega)) \leq u\} = \{\omega : X(\omega) \leq \tau(u)\}.$$

Thus

$$P(\{\omega : Z(\omega) \leq u\}) = F(\tau(u)) = u$$

and, therefore, Z is uniformly distributed on $(0, 1)$. ■

B. Lemmas for Theorem 2

The following two lemmas will be used in proving Theorem 2.

Lemma 2: If a function $\phi(x, y) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ is a continuous function in x (for fixed y) and a Borel function in y (for fixed x), then it is a Borel function.

Proof: The set $\{(x, y) : a < \phi(x, y) < b\}$ can be expressed as a countable union of sets, each of which is a cross product of an interval in x with a countable intersection of measurable sets in y . ■

Lemma 3: Let $g(y) : \mathbb{R} \rightarrow \mathbb{R}$ be a Borel function. Then, from definition (1), we have

$$\int_{\{\omega : Y(\omega) \in B\}} I_{A(g(Y(\omega)))} P(d\omega) = \int_B m_{g(y)}(y) P_Y(dy).$$

Proof: The above is true for $Y(\omega)$ as a simple random variable. The general case then follows. ■

C. Proof of Theorem 2

Proof: Measurability of $Z(\omega)$ follows from Lemma 1. For $y \in \mathbb{R}$ and $u \in (0, 1)$, define

$$\tau(u, y) \triangleq \sup\{x \in \mathbb{R} : M(x, y) \leq u\}. \quad (8)$$

Since $M(\cdot, y)$ is continuous, we have

$$M(\tau(u, y), y) = u. \quad (9)$$

Define

$$S \triangleq \{(x, y) \in \mathbb{R} \times \mathbb{R} : M(x, y) \leq u\}. \quad (10)$$

Combining (8) and (9) with the fact that $M(\cdot, y)$ is nondecreasing, we have

$$S = \{(x, y) \in \mathbb{R} \times \mathbb{R} : x \leq \tau(u, y)\}. \quad (11)$$

Let $I_S(X(\omega), Y(\omega))$ be the indicator function for the set $\{\omega \in \Omega : (X(\omega), Y(\omega)) \in S\}$. From (10) and (11), it follows that

$$\begin{aligned} \int_{\{\omega : Y(\omega) \in B\}} I_S(X(\omega), Y(\omega)) P(d\omega) \\ = \int_{\{\omega : Y(\omega) \in B\}} I_{A(\tau(u, Y(\omega)))} P(d\omega). \end{aligned}$$

From Lemma 2, we can transform the right hand side of the above, as follows:

$$\int_{\{\omega : Y(\omega) \in B\}} I_{A(\tau(u, Y(\omega)))} P(d\omega) = \int_B M(\tau(u, y), y) P_Y(dy).$$

Thus, we can define the conditional expectation

$$E(I_S(X(\omega), Y(\omega)) | Y(\omega) \triangleq y) \triangleq M(\tau(u, y), y) = u.$$

Now we have for $u \in (0, 1)$

$$\begin{aligned} P(Z \leq u) &= P((X, Y) \in S) \\ &= E(I_S(X(\omega), Y(\omega))) \\ &= E(E(I_S(X(\omega), Y(\omega)) | Y(\omega))) \\ &= \int_B E(I_S(X(\omega), Y(\omega)) | Y(\omega) \triangleq y) P_Y(dy) \\ &= \int_B u P_Y(dy) \\ &= u \end{aligned}$$

and, therefore, Z is uniformly distributed on $(0, 1)$. ■

ACKNOWLEDGMENT

L. Chen would like to thank Dr. A. Zatezalo, Dr. J. Cabrera, and Dr. R. Prasanth for many helpful discussions, comments, and critical review of the draft. The authors would like to thank Prof. A. Budhiraja for many insightful discussions which greatly contributed to the formation of the idea and the writing of this technical note. We are also grateful for his review of the draft and valuable feedback. They would also like to thank the Associate Editor and reviewers for their valuable comments and helpful suggestions.

REFERENCES

- [1] I. Ishida, "Scanning multivariate conditional densities with probability integral transforms," CIRJE, Faculty of Economics, University of Tokyo, CIRJE F-Series CIRJE-F-369, Sep. 2005 [Online]. Available: <http://ideas.repec.org/p/ky/fseries/2005cf369.html>
- [2] A. N. Shiryaev, *Probability*. New York: Springer, 1995.
- [3] NIST/SEMATECH e-handbook of statistical methods [Online]. Available: <http://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm>
- [4] M. Rosenblatt, "Remarks on a multivariate transformation," *Ann. Math. Statist.*, vol. 23, no. 3, pp. 470–472, 1952.
- [5] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
- [6] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.
- [7] J. Angus, "The probability integral transform and related results," *SIAM Rev.*, vol. 36, no. 4, pp. 652–654, Dec. 1994.

On Improving Transient Performance in Tracking Control for a Class of Nonlinear Discrete-Time Systems With Input Saturation

Yingjie He, Ben M. Chen, and Weiyao Lan

Abstract—Quick response and small overshoot are two desired transient performances of target tracking control. While most of the design schemes compromise between these two performances, we try to achieve both simultaneously for the tracking control of a class of nonlinear discrete-time systems with input saturation by using a composite nonlinear feedback (CNF) control technique. The closed-loop system with improved transient performance preserves the stability of the nonlinear part of the partially linear composite system.

Index Terms—Discrete-time systems, input saturation, nonlinear systems, tracking control, transient performance.

I. INTRODUCTION

The tracking control problems, such as target tracking [4] and output regulation [8], are extensively studied in the literature. Settling time

Manuscript received February 2, 2005; revised November 7, 2005, June 12, 2006, and March 7, 2007. Recommended by Associate Editor D. Nesic.

Y. He was with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore. He is now with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore.

B. M. Chen is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore (e-mail: bm-chen@nus.edu.sg).

W. Lan was with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore. He is now with the Department of Automation, Xiamen University, Xiamen, Fujian 361005, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2007.900836

and overshoot are two important transient performance indices, and quick response and small overshoot are desirable in the most of the target tracking control problems. However, it is well known that, in general, quick response results in a large overshoot. Thus, most of the design schemes have to make a tradeoff between these two transient performance indices. To improve the transient performance, Lin *et al.* [14] proposed a composite nonlinear feedback (CNF) control technique for a class of second-order linear systems. The CNF control law consists of a linear feedback law and a nonlinear feedback law without any switching element. The linear feedback part is designed to yield a closed-loop system with a small damping rate for quick response, while the nonlinear feedback part is used to increase the damping ratio of the closed-loop system as the system output approaches the target reference to reduce the overshoot. Turner *et al.* [22] later extended the results of [14] to higher order and multiple-input systems under a restrictive assumption on the system. However, both [14] and [22] considered only the state feedback case. Recently, Chen *et al.* [2] have developed a CNF control design to a more general class of systems with measurement feedback, and successfully applied the technique to solve a hard-disk servo problem. The extension of this idea to general linear continuous multiple-input–multiple-output (MIMO) systems is found in [6]. The CNF control techniques for linear discrete-time systems can be found in [7] and [23].

This note aims to design a CNF control law for discrete-time partially linear composite systems with input saturation. The results for its continuous-time counterpart have been reported in [12]. In the last two decades, the nonlinear control problems for partially linear composite systems have been extensively studied by many researchers such as [10], [15], [19], and [20], to name just a few. It was shown in [19] that a nonlinear system which is zero-input globally asymptotically stable (GAS) will preserve its GAS property if its input decreases to zero with a very fast exponential rate. However, a bad transient performance may destroy the stability of the nonlinear part before the output rapidly decays to zero. This is also true for discrete-time systems since the inter-sampling behavior is equivalent to the response of a continuous-time system with unchanging input. Based on the linear part of the composite system, the CNF control is designed such that the closed-loop system has desired performances, e.g., quick response and small overshoot. Moreover, we show that the closed-loop system with improved transient performance preserves the stability of the nonlinear part of the partially linear composite system.

II. PROBLEM FORMULATION

Consider a partially linear composite discrete-time systems with input saturation Σ characterized by

$$\xi(k+1) = f(\xi(k), x(k), y(k)), \quad \xi(0) = \xi_0 \quad (1)$$

$$x(k+1) = Ax(k) + Bs\text{at}(u(k)), \quad x(0) = x_0 \quad (2)$$

$$y(k) = Cx(k) \quad (3)$$

where $(\xi, x) \in \mathbb{R}^m \times \mathbb{R}^n$, $u \in \mathbb{R}$ and $y \in \mathbb{R}$ are, respectively, the state, control input, and control output of the given system Σ , f is a continuous function, A , B , and C are appropriate dimensional constant matrices, and the saturation function is defined by

$$\text{sat}(u) = \text{sign}(u)\min(|u|, u_{\max}) \quad (4)$$

where u_{\max} is the maximum amplitude of the control channel.