

An adaptive particle filter based on PSO and fuzzy inference system for nonlinear state systems

Ramazan Havangi

To cite this article: Ramazan Havangi (2018) An adaptive particle filter based on PSO and fuzzy inference system for nonlinear state systems, *Automatika*, 59:1, 94-103, DOI: [10.1080/00051144.2018.1498207](https://doi.org/10.1080/00051144.2018.1498207)

To link to this article: <https://doi.org/10.1080/00051144.2018.1498207>



© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 03 Aug 2018.



Submit your article to this journal [↗](#)



Article views: 154



View related articles [↗](#)



View Crossmark data [↗](#)



An adaptive particle filter based on PSO and fuzzy inference system for nonlinear state systems

Ramazan Havangi

Faculty of Electrical and Computer Engineering, University of Birjand, Birjand, Iran

ABSTRACT

Particle filters have been widely used in nonlinear/non-Gaussian Bayesian state estimation problems. However, the particle filter (PF) is inconsistent over time. The inconsistency of PF mainly results from the particle depletion in resampling step and an incorrect priori knowledge of process and measurement noise. To cope with this problem and enhance the accuracy and consistency of the state estimation, an adaptive particle filter (APF) is proposed in this paper. In APF, an adaptive fuzzy square-root unscented Kalman filter (AFSRUKF) is used to generate the proposal distribution. This causes that beside the merit of reducing the computational cost, APF has some other advantages such as increasing consistency that leads to more numerical stability and better performance. Moreover, APF can work in unknown statistical noise behaviour and is more robust. This is why the fuzzy inference system (FIS) supervises the performance of square-root unscented particle filter (SRUPF) using tuning statistical noises. In APF, to increase the diversity of particles, the resampling process is done based on the particle swarm optimization (PSO). With this resampling strategy, the small-weight particles are modified to the large-weight ones without duplication and elimination of particles. The effectiveness of APF is demonstrated by using two experiment examples through Monte Carlo simulations. The simulation results demonstrate the effectiveness of the proposed method.

ARTICLE HISTORY

Received 26 September 2016
Accepted 5 June 2018

KEYWORDS

Fuzzy inference system;
particle filter; particle swarm
optimization (PSO)

Introduction

Nonlinear filtering problems arise in many fields such as target tracking [1,2], robot navigation [3,4], etc. It is a process of finding a state estimate of a nonlinear system based on a set of noisy measurements [5,6]. The Kalman filter (KF) is a commonly used state estimation technique for the linear systems with Gaussian noise [6].

To extend KF to the nonlinear system with Gaussian noise, modified KFs such as extended KF (EKF) and unscented KF (UKF), have been proposed [7–9]. However, for the system with high nonlinearity, the poor state estimation results will be obtained [5,6].

Particle filter is as an effective estimator for the nonlinear filtering problem and can be easily applied to nonlinear/non-Gaussian dynamic models that may be difficult for other filtering techniques [5,6]. The PF is a Monte Carlo method for implementing recursive Bayesian estimation that constructs probability density function (PDF) using a set of random particles with associated weights [5,6]. In addition, in PF, the particles are evolved over time via a combination of importance sampling and resampling step. In comparison with KFs, PF can provide more accurate state estimation results, especially for the highly nonlinear systems [5].

However, there are some serious problems encountered in the general PF [8,9]. One of them is particle impoverishment. Most of the particles share a few distinct values and the posterior distribution is insufficiently approximated [9,10]. This problem can lead to the misleading state estimation results. A common way to overcome the particle impoverishment problem is selecting a good proposal distribution and resampling scheme. In general PF, the state transition is often chosen as the proposal distribution [10,11]. As this proposal distribution does not include information of the new observations, the most particles get negligible weights and it leads to particle degeneracy [12–14]. By selecting a good the proposal distribution, which contains the current measurement information, the particle impoverishment problem can also be alleviated.

Following this idea, the extended Kalman particle filter (EPF) is proposed in [15,16], which uses EKF to generate proposal distribution. In [17], the EKF proposal is replaced by a UKF proposal and the unscented particle filter (UPF) is proposed.

However, a serious limitation of UPF is that it requires the complete a priori knowledge of the noise statistics that in most practical applications are

unknown [18,19]. Evidences have shown how a poor estimation of the input noise statistics may seriously degrade the performance of UKF and even induce the divergence of it [20–22]. In addition, in UPF, an inherent rounding error in the numerical calculation may cause negative definiteness of the state covariance, which will lead to filter divergence and influence the estimation precision [23–25]. Finally, the resampling step in UPF leads to a great loss of diversity in particles [26,27].

In this paper, APF is proposed to overcome these problems. The proposed method uses AFSRUKF instead of UKF to generate the proposal distribution. Beside the merit of reducing the computational cost, APF has some other advantages such as increasing consistency that leads to more numerical stability and better performance. This is mainly because of the fact that all resulting covariance matrices are guaranteed to remain positive semi-definite. In addition, APF can work in unknown statistical noise behaviour and thus it is more robust. The fuzzy inference system (FIS) supervises the performance of SRUPF with tuning statistical noises (R_t and Q_t) to close theoretical covariance to actual covariance. This adaptive tuning provides more robustness and consistency for the filter, which leads to results that are more accurate.

In order to reduce the impact of resampling on the accuracy and consistency, the particle swarm optimization (PSO)-based resampling scheme is used in APF. In the PSO resampling scheme, the particles recombine by using an iterative process and selection and unlike the traditional resampling schemes; there is no duplication and elimination of particles. The new resampling algorithm can maintain the diversity of particles, and ensure the resampled particles asymptotically approximate the samples PDF of the true state.

The rest of the paper is organized as follows. In Section 2, the required background is reviewed. The adaptive particle filter is presented in Section 3. In Section 4, the results are compared and analysed with simulation results.

Background

Particle filter

Consider the following nonlinear state space model:

$$\begin{aligned} x_t &= f(x_{t-1}) + w_{t-1}, \\ y_t &= h(x_t) + v_t, \end{aligned} \quad (1)$$

where $f(\cdot), h(\cdot)$ are known nonlinear functions with appropriate dimensions, $x_t \in R^n$ is the state vector, $y_t \in R^m$ is the measurement vector, w_{t-1} , v_t are independent white-noise variables. The objective of filtering is to estimate the posterior density of the states given the past measurements $p(x_t|y_{1:t})$ [5,6]. In PF, the posterior density can be calculated by a set of weighted particles

drawn as follows:

$$p(x_t|y_{1:t}) = \sum_{i=1}^N \omega_t^{(i)} \delta(x_t - x_t^{(i)}), \quad (2)$$

where $\delta(x_t - x_t^{(i)})$ is Dirac's delta function, and N is the number of particles. Unfortunately, the $p(x_t|y_{1:t})$ is often unknown, so actually, it is impossible to sample directly from the $p(x_t|y_{1:t})$. Aiming to the problem, PF circumvents the difficulty by sampling from known, easy-to-sample, the proposal distribution $q(x_t|y_{1:t})$ which should be as far as possible close to $p(x_t|y_{1:t})$. The associated importance weight of the particle is defined as follows [5]:

$$\omega_t^{(i)} = \frac{p(x_t^{(i)}|y_{1:t})}{q(x_t^{(i)}|y_{1:t})}. \quad (3)$$

The resampling step is performed to avoid degeneracy in the particle filter.

Particle swarm optimization

The particle swarm algorithm is a population-based metaheuristic optimization method that is based on the simulation of the social behaviour of birds within a flock [30,31]. In PSO, the potential solutions of the optimization problem that are called particles, move in the search space with a velocity, which is adjusted as a function of the position of other particles. All the particles follow the best particle during the iterations and converge together to the global optimum of the considered objective function. Suppose that the search space dimension is D , and the number of particles is NP , the position and velocity of the i -th particle are represented by $x^{(i)} = [x_{i1}, \dots, x_{iD}]$ and $v^{(i)} = [v_1, \dots, v_{iD}]$, respectively.

Let $P_{bi} = [p_{i1}, \dots, p_{iD}]$ denotes the best position that particle i is achieved so far, and P_g denote the best position of P_{bi} for any $i = 1, \dots, NP$. The PSO algorithm may be performed by the following formulations [30,31]:

$$x_k^{(i)} = x_{k-1}^{(i)} + v_k^{(i)}, \quad (4)$$

$$v_k^{(i)} = wv_{k-1}^{(i)} + c_1r_1(P_{bi} - x_{k-1}^{(i)}) + c_2r_2(P_g - x_{k-1}^{(i)}), \quad (5)$$

where k represents the iteration number and c_1, c_2 are positive coefficients. Usually $c_1 = c_2 = 2$, and r_1, r_2 are random numbers in the interval $(0, 1)$, w is the inertial weight.

Adaptive particle filter

In this section, the adaptive particle filter is presented to overcome the drawbacks of PF. The proposed algorithm consists of sampling, importance weighting and PSO resampling. In the following subsections, details of these steps are given.

Sampling

The choice of proposal distribution is one of the most critical issues in the design of a particle filter [5,6]. In APF, the proposal distribution is as follows:

$$x_t^{(i)} \sim q(x_t^{(i)} | x_{t-1}^{(i)}, y_t) = N(x_t; x_{t|t}^{(i)}, P_{t|t}^{(i)}), \quad (6)$$

where $N(x_t; x_{t|t}^{(i)}, P_{t|t}^{(i)})$ is the Gaussian distribution (with mean $x_{t|t}^{(i)}$ and covariance $P_{t|t}^{(i)}$) obtained by using SRUKF. The SRUKF is based on the idea that the square root of a matrix has a better condition number than that of the original number and its use reduces numerical errors [24,25]. In the proposed method, the square-root matrix $S_{t|t}$ of $P_{t|t}$ is updated, where $P_{t|t} = S_{t|t} S_{t|t}^T$. In addition, the state update and Kalman gain equation use $S_{t|t}$ instead of $P_{t|t}$. This is particularly useful since the unscented transform (UT) uses the square root $S_{t|t}$ of $P_{t|t}$ and propagating $S_{t|t}$ eliminates the Cholesky factorization. Let $x_{t-1|t-1}$ be the estimated state and $S_{t-1|t-1}$ be the square root of $P_{t-1|t-1}$ based on the available measurements up to current time instant $t - 1$. To draw samples from above proposal distribution, a set of $2n + 1$ points is generated by

$$\begin{aligned} \chi_{t-1|t-1}^{(0)(i)} &= x_{t-1|t-1}^{(i)}, \\ \chi_{t-1|t-1}^{(j)(i)} &= x_{t-1|t-1}^{(i)} + \gamma(S_{t-1|t-1}^{(i)})_j \quad j = 1, \dots, n_x, \\ \chi_{t-1|t-1}^{(j)(i)} &= x_{t-1|t-1}^{(i)} - \gamma(S_{t-1|t-1}^{(i)})_j \quad j = n + 1, \dots, 2n_x, \\ S_{t-1|t-1}^{(i)} &= \text{chol}(P_{t-1|t-1}^{(i)}), \end{aligned} \quad (7)$$

where $(S_{t-1|t-1}^{(i)})_i$ means the i -th column of the square root of the covariance matrix. The sigma propagated through the nonlinear system and measurement models as

$$\begin{aligned} \bar{x}_{t|t-1}^{-(j)(i)} &= f(\chi_{t-1|t-1}^{(j)(i)}), \\ \bar{\mathfrak{S}}_t^{(j)(i)} &= h(\bar{x}_{t|t-1}^{-(j)(i)}), \end{aligned} \quad (8)$$

where $\bar{x}_{t|t-1}^{-(j)(i)}$ is its transformed sigma point. The prediction of the measurement vector $\bar{y}_t^{(i)}$, prediction of mean $x_{t|t-1}^{(i)}$ and square-root covariance matrix $S_{t|t-1}^{[m]}$ of the state are as follows:

$$\bar{y}_t^{(i)} = \sum_{j=0}^{2n} w_m^{(j)} \bar{\mathfrak{S}}_t^{(j)(i)}, \quad (9)$$

$$x_{t|t-1}^{(i)} = \sum_{j=0}^{2n} \omega_m^{(j)} \bar{x}_{t|t-1}^{-(j)(i)}, \quad (10)$$

$$\begin{aligned} S_{t|t-1}^{(i)} &= \text{qr} \left\{ \left[\sqrt{\omega_c^{[1]}} (\bar{x}_t^{(1:2n)(i)} - x_{t|t-1}^{(i)}) \quad \sqrt{Q_{t-1}} \right] \right\}, \\ S_{t|t-1}^{(i)} &= \text{cholupdate}\{S_{t|t-1}^{(i)}, (\bar{x}_t^{(0)(i)} - x_{t|t-1}^{(i)}), \omega_c^{(0)}\}, \end{aligned} \quad (11)$$

where the weights ω_m and ω_c are as

$$\begin{aligned} \omega_m^{(0)} &= \frac{\lambda}{(n + \lambda)}, \quad \omega_c^{(0)} = \frac{\lambda}{(n + \lambda)} + (1 - \alpha^2 + \beta), \\ \omega_m^{(j)} &= \omega_c^{(j)} = \frac{\lambda}{2(n + \lambda)} \quad (j = 1, \dots, 2n). \end{aligned} \quad (12)$$

Finally, the state mean $x_{t|t}^{(i)}$ and square root of the covariance matrix $S_{t|t}^{(i)}$ are updated as

$$x_{t|t}^{(i)} = x_{t|t-1}^{(i)} + K_t^{(i)}(y_t - \bar{y}_t^{(i)}), \quad (13)$$

$$U = K_t^{(i)} S_{y_t}, \quad (14)$$

$$S_{t|t}^{(i)} = \text{cholupdate}\{S_{t|t-1}^{(i)}, U, -1\},$$

where S_{y_t} is square and triangular, $h(\cdot)$ is the observation model and $K_t^{(i)}$ is the gain matrix:

$$K_t^{(i)} = (P_{\mathfrak{S}_v}^{(i)} / S_{y_t}^T) / S_{y_t}, \quad (15)$$

and cross covariance $P_{\delta v}^{(i)}$ is as follows:

$$P_{\delta v}^{(i)} = \sum_{j=0}^{2n} \omega_c^{(j)} (\bar{x}_{t|t-1}^{(j)(i)} - x_{t|t-1}^{(i)}) (\bar{\mathfrak{S}}_t^{(j)(i)} - \bar{y}_t^{(i)})^T, \quad (16)$$

$$S_{y_t} = \text{qr} \left\{ \left[w_c^{(1)} (\bar{\mathfrak{S}}_t^{(1:2n)(i)} - \bar{y}_t^{(i)}), \sqrt{R_t} \right] \right\}, \quad (17)$$

$$S_{y_t} = \text{cholupdate}\{S_{y_t}, (\bar{\mathfrak{S}}_t^{(0)(i)} - \bar{y}_t^{(i)}), w_c^{(0)}\}.$$

From the Gaussian distribution generated by the estimated mean and covariance of the vehicle, the state of each particle is sampled as

$$x_t^{(i)} \sim N(x_t; x_{t|t}^{(i)}, P_{t|t}^{(i)}), \quad (18)$$

where the covariance matrix of the robot pose is as follows:

$$P_{t|t}^{(i)} = (S_{t|t}^{(i)})^T S_{t|t}^{(i)}. \quad (19)$$

From the above equations, it can be seen that the covariance matrices R_t and Q_t will affect the precision of the SRUPF directly. In practical systems, the priori knowledge of the process noise and measurement noise is unknown. This is why it is hard to determine process noise and measurement noise covariances. The use of wrong priori statistics in SRUPF may lead to large estimation errors or even to the divergence. So the accuracy of R_t and Q_t has become the key problem which affects the precision and stability of the filter and needs to be tuned online. In this paper, an online adaptive scheme of SRUPF is presented. The adaptation is in the sense of adaptively adjusting the process noise and measurement noise covariances using the covariance-matching technique.

The basic idea behind the covariance-matching technique is to make the actual value of the covariance of the innovation sequence match its theoretical value [20–22]. If the theoretical covariance and its actual covariance are discrepant, FIS will adjust R_t and Q_t online in order to reduce the discrepancy. The adjusting factor is generated by FIS based on the degree of the discrepancy. Assuming the noise covariance matrix Q_t is known, the adaptive adjustment of the matrix R_t is done. For this purpose, the innovation sequence denotes as follows:

$$v_t = y_t - \hat{y}_t.$$

From the measurement equation of UKF, the theoretical covariance of v_t is computed by

$$\begin{aligned} S_t &= E(v_t v_t^T) \\ &= \sum_{j=0}^{2n} w_c^{(j)} (\mathfrak{S}_t^{(j(i))} - \bar{y}_t^{(i)}) (\mathfrak{S}_t^{(j(i))} - \bar{y}_t^{(i)})^T + R_t. \end{aligned} \quad (20)$$

The actual covariance of v_t can be approximated as

$$C_t = \frac{1}{m} \sum_{k=0}^{m-1} v_k v_k^T,$$

where m is the width of the moving window and is chosen empirically to give some statistical smoothing, v is the residual vector. If the window size is too small, the measurement estimation covariance can be noisy.

In general, the window size is chosen empirically for statistical smoothing [21–22]. It should be pointed out that too large a window width suffers from the severe computational inefficiency and too small a window width might lead to large variance. Thus, it is important to choose a reasonable window width. If a discrepancy is found between the actual value and the theoretical value, then R_t is tuned based on the knowledge of the degree of the discrepancy using FIS. The tuning process is shown in Figure 1. Define the discrepancy between the actual covariance C_t and the theoretical covariance S_t as DOM_t (Degree of Matching) as follows:

$$DOM_t = S_t - C_t.$$

If the actual covariance is less than its theoretical then the value of R_t should be increased. If the actual value

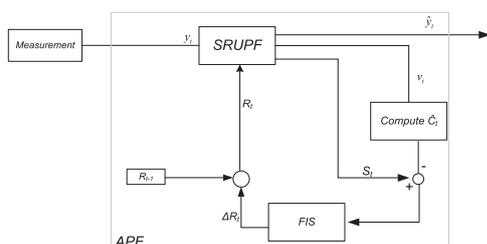


Figure 1. Tuning SRUKF in APF using FIS.

of covariance C_t is greater than its theoretical value S_t , then the value of R_t should be decreased. Thus, the DOM_t can be used to reduce the discrepancy between S_t and C_t . To implement the above covariance-matching technique using FIS, the general rules of adaptation are as follows:

If $DOM_t(k, k) \cong 0$ then maintain $R_t(k, k)$ unchanged

If $DOM_t(k, k) > 0$ then decrease $R_t(k, k)$

If $DOM_t(k, k) < 0$ then increase $R_t(k, k)$

Adjustment of R_t is performed using FIS is as the following relation:

$$R_t(k, k) = R_{t-1}(k, k) + \Delta R_t(k, k).$$

Figure 1 shows tuning SRUPF using FIS where ΔR_t is the FIS output, DOM_t is the FIS input. The membership function for DOM_t and ΔR_t are shown in Figures 2 and 3.

Importance weighting

The importance weight of each particle is computed through Equation (21) when the observation at time t is obtained:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)})}{N(x_t; x_{t|t}^{(i)}, P_{t|t}^{(i)})}, \quad (21)$$

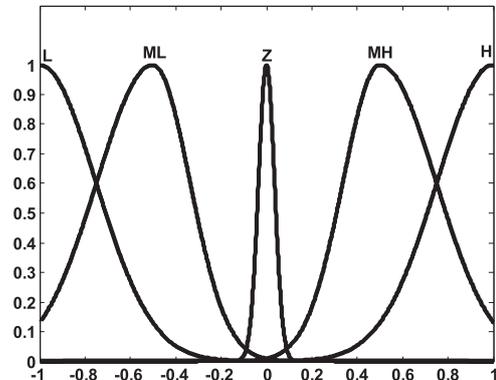


Figure 2. Membership function DOM_t .

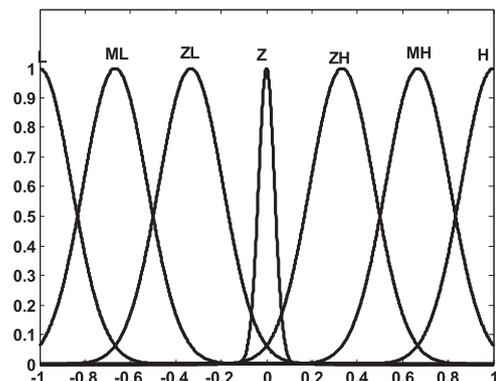


Figure 3. Membership function ΔR_t .

then, the particle weight is normalized as follows:

$$w_t^{(i)} = \frac{w_t^{(i)}}{\sum_{k=1}^N w_t^{(k)}}.$$

PSO resampling

It is known that the objective of resampling steps is to eliminate particle degeneracy problem [26–28]. The idea of resampling is that particles with large weights are duplicated while those with small weights are abandoned. The common resampling techniques are residual, multinomial, systematic and stratified [27–29]. These algorithms use weight strata to decide how many copies of each particle should be made. Although these techniques have good performance, they inevitably lead to the loss of diversity among the particles. In this paper, in order to restrain losing diversity, an effective resampling scheme based on the PSO (PSO resampling) is introduced. The PSO resampling is performed when the effective number of samples reduces below a threshold N_{eff} , which is calculated as

$$N_{eff} = \frac{1}{\sum_{k=1}^N (\omega_t^{(k)})^2}. \quad (22)$$

In the PSO resampling scheme, the particles $\{x_t^{(i)}; i = 1, \dots, N\}$ are regarded as the target vectors of the current population, and the corresponding weights $\{\omega_t^{(i)}; i = 1, \dots, N\}$ are as the objective functions of the target vectors, respectively. Unlike the traditional resampling schemes, there is no elimination and replication of particles. In fact, the PSO resampling recombines the particles by using an iterative process. Let $C_{t,k} = \{x_{t,k}^{(1)}, \dots, x_{t,k}^{(N)}\}$ denote the current population of PSO, which consists of N D-dimensional target vectors. where $x_{t,i}^k$ represents each candidate solution i to the optimization problem at iteration k for a given time step t .

The PSO algorithm updates the velocity and position of each particle as follows [30]:

$$x_{t,k}^{(i)} = x_{t,k-1}^{(i)} + v_{t,k}^{(i)}, \quad (23)$$

$$v_{t,k}^{(i)} = wv_{t,k-1}^{(i)} + c_1r_1(P_{bi} - x_{t,k-1}^{(i)}) + c_2r_2(P_g - x_{t,k-1}^{(i)}). \quad (24)$$

The fitness function of $x_{t,k}^{(i)}$ is calculated as

$$f(x_{t,k}^{(i)}) = \frac{p(y_t^{(i)} | x_{t,k}^{(i)})p(x_{t,k}^{(i)} | x_{t,k-1}^{(i)})}{N(x_t; x_t^{(i)}, P_t^{(i)})}. \quad (25)$$

PSO moves all particles towards the particle with the best fitness function. When the best fitness value

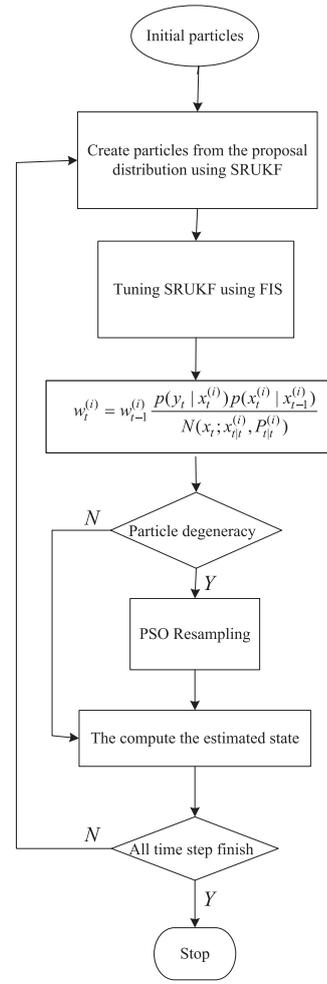


Figure 4. Adaptive particle filter.

reaches a certain threshold, the iteration is stopped. At the end of iterations, P_{gbest} and P_{pbest} are calculated and the particles are distributed according to the values of P_{gbest} and P_{pbest} . With this set of particles, the sampling process will be conducted on the basis of proposal distribution. In summary, a single iteration of the Adaptive particle filter is given in Figure 4.

Results

In order to verify the performance of APF, the performance of it is evaluated and compared with UPF, PF, MCMC-based particle filtering and EPF on simulated data under different condition. Two examples are investigated in this section. The first example is a univariate growth model, which is popular in econometrics and has been used previously in [11]. The second example is robot localization, which is of interest in robot applications.

Univariate growth model

This model is a benchmark model that is commonly used in PF testing. This system has been used before in many papers [5–6]. The state space model of this system

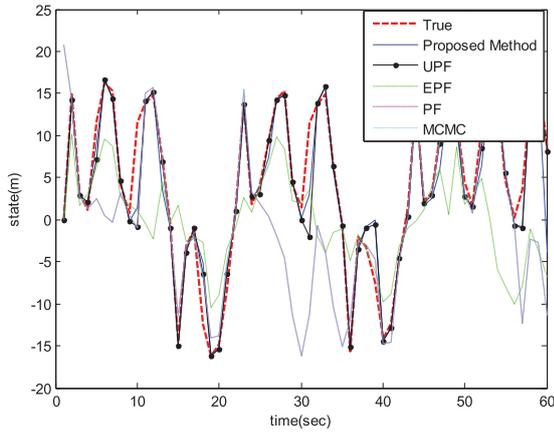


Figure 5. The state estimation results.

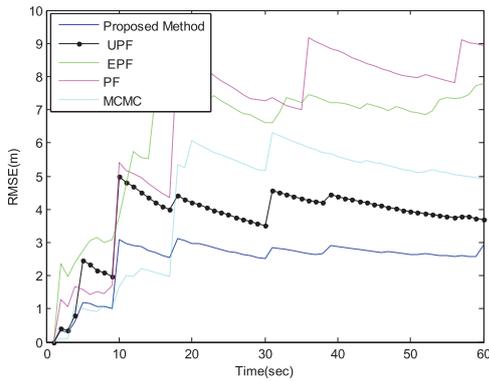


Figure 6. RMSE of algorithms with respect to time.

is as follows:

$$\begin{aligned} x_t &= \frac{1}{2}x_{t-1} + 25 \frac{x_{t-1}}{1+x_{t-1}^2} + 8 \cos(1.2t) + \omega_{t-1}, \\ y_t &= \frac{x_t^2}{20} + v_t, \end{aligned} \quad (27)$$

where the initial state is $x_0 = 0.1$, the covariance of the measurement noise is $R_t = 1$, and the covariance of the process noise is $Q_{t-1} = 1$. Fifty particles (i.e. $N = 50$) are used in all algorithms.

At first, the performance of the proposed method is compared with UPF, EPF, PF and MCMC-based particle filtering when the statistical properties of noises are known a priori. In Figure 5, the tracking performance of the algorithms is depicted. It can be observed that the best state estimation results belong to the proposed method.

To evaluate the estimations accuracy, the root mean square error (RMSE) is used to evaluate each algorithm. The RMSE is obtained over 100 runs. The RMSE of estimations over time is shown in Figure 6. It can be seen that APF provides a more accurate estimation than the EPF and UPF. The reason is that PSO resampling can obviously increase the number of meaningful particles and enhance the performance of estimation. As result, it can enrich the particle species and capture the distribution more comprehensively and accurately. The number of distinct particles is shown in Figure 7. As can be seen,

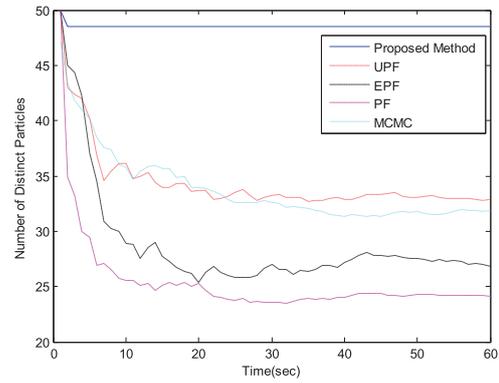


Figure 7. The number of distinct particles.

Table 1. The performance of algorithms with the number of particles.

Number of particles	RMSE			Processing time(sec)		
	APF	UPF	EPF	APF	UPF	EPF
100	2.9	3.4	5.2	5.4	4.23	2.8
70	3.1	3.5	6.2	4.3	3.2	2
50	3	3.7	7.8	3.5	2.73	1.54
30	3.3	6	10	2.8	1.69	0.96
10	3.6	10	12	0.95	0.65	0.36

the number of distinct particles of the proposed method is more than that of other algorithms. As a result, the consistency of APF is increased.

To examine the sensitivity of the performance of algorithms with the number of particles, three different particle numbers of 100, 70, 50, 30, and 10, are used for simulation tests. Table 1 shows the performance of algorithms with the number of particles. As can be seen, in general, when the more particles are used, the estimation accuracy of the algorithms is higher. However, the APF gives the least averaged variance with different particles. The Table 1 shows also the computational cost of algorithms. As can be observed, the smallest computation time belongs to EPF. However, the APF requires less number of particles to obtain the same estimation accuracy compare to EPF and UPF.

Second, the performance of algorithms is evaluated when the statistical properties of noises are unknown. In all algorithms, the number of particles is 50. The results are shown in Figures 8 and 9. Compared with the proposed method, the state estimate of UPF, MCMC, EPF and PF deviated from the true state. From a comparison of Figures 6 and 9, it is easy to conclude that the performance of APF, in this case, is almost close to the previous case, while the performance of other methods is worse than that of them in the previous case. The reason is that the APF can estimate the unknown measurement noises online whereas other method depends on the fixed prior knowledge about the measurement noises.

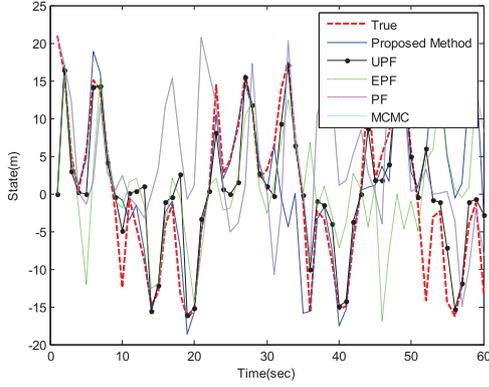


Figure 8. The state estimation results.

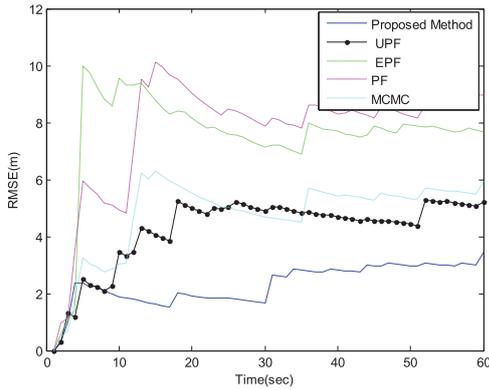


Figure 9. RMSE of algorithms with respect to time.

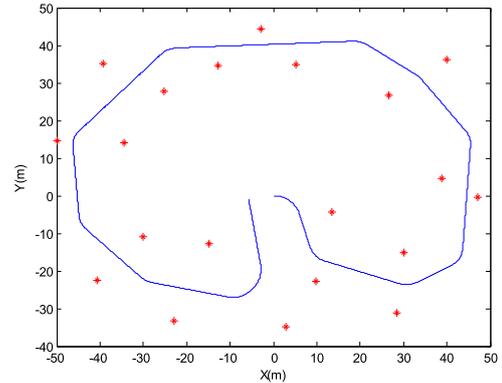
Robot localization

In this example, the performance of the proposed algorithm in robot localization is studied. The robot localization is one of many issues in mobile robot study. Localization is an essential ability for a mobile robot to determine its pose (location and orientation angle) from sensor data so that it can plan a movement and go to the desired location. Suppose the kinematic model of the mobile robot is as follows:

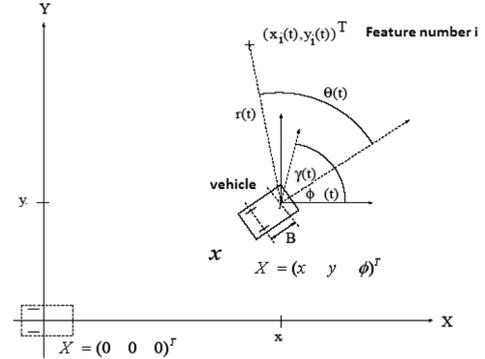
$$\begin{bmatrix} x_t \\ y_t \\ \phi_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + \Delta t(V + v_v)\cos(\varphi + \gamma + v_\gamma) \\ y_{t-1} + \Delta t(V + v_v)\sin(\varphi + \gamma + v_\gamma) \\ \varphi_{t-1} + \Delta t \frac{(V + v_v)}{B} \sin(\gamma + v_\gamma) \end{bmatrix}, \quad (28)$$

where (x, y) are the location robot, ϕ is the orientation robot respectively to the global environment, Δt is the time interval, B is the base line of the robot, $u = [V \ \gamma]^T$ is the control input at time t consisting of a velocity input V and a steer input γ , $v = [v_v \ v_\gamma]^T$ is the process noise. The robot is assumed to be equipped with a range-bearing sensor that provides a measurement of range r_i and bearing θ_i to an observed feature ρ_i relative to the robot. The observation z of feature ρ_i in the map can be expressed as [32]

$$z = \begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x - x_i)^2 + (y - y_i)^2} + \omega_r \\ \tan^{-1} \frac{y - y_i}{x - x_i} - \phi + \omega_\theta \end{bmatrix}, \quad (29)$$



(a)



(b)

Figure 10. (a) Experiment environment of robot (b) wheel mobile robot schematic.

where (x_i, y_i) is the landmark position in the map, and $\omega = [\omega_r \ \omega_\theta]^T$ is measurement noise. The robot localization based on the proposed algorithm is tested for experiment environment as shown in Figure 10. The star points (*) depicts the location of the landmarks that are known and stationary in the environment and the blue line is the path of the robot.

The initial position of the robot is assumed to be $(0, 0, 0)$ and the robot moves at a speed of 3 m/s and with a maximum steering angle of 30° . The robot has 4 m wheel base and is equipped with a range-bearing sensor with a maximum range of 30 m and 180° frontal field-of-view. The control noise is $\sigma_v = 0.3$ m/s, $\sigma_\gamma = 3^\circ$ and the measurement noise is 0.1 m in range and 0.1° in bearing. A control frequency is 40 Hz and observation scans are obtained at 5 Hz. Figures 11–14 show the comparison of the RMSE of the localization based on proposed algorithm and UPF when the noise statistics are known and unknown to both algorithms. The results are obtained over 100 Monte Carlo runs with 20 particles.

As can be seen, the localization based on APF outperforms all the other counterparts in both the robot position and heading errors. This is because the localization based on APF not only improves the numerical stability but also reduce the negative effects caused by the resampling step. In addition, APF can estimate the unknown measurement noises online.

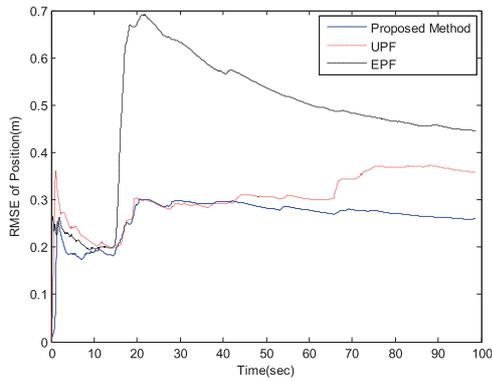


Figure 11. RMSE of robot position when the statistics of the noises are assumed to be known.

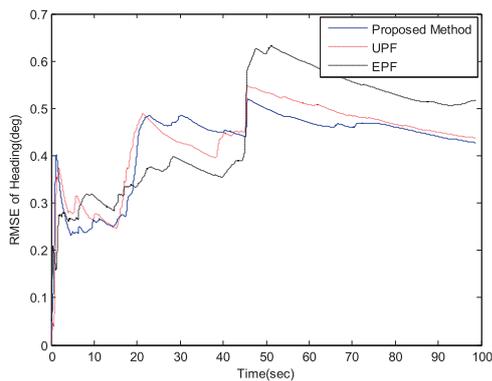


Figure 12. RMSE of heading of robot when the statistics of the noises are assumed to be known.

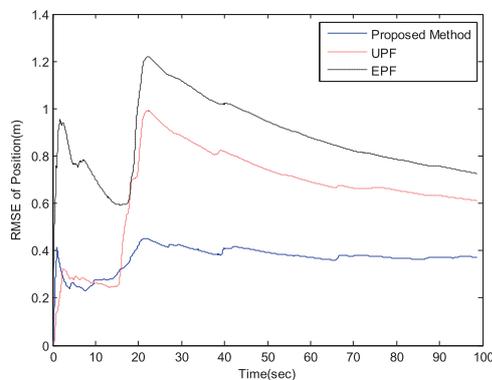


Figure 13. RMSE of robot position when the statistics of the noises are assumed to be unknown.

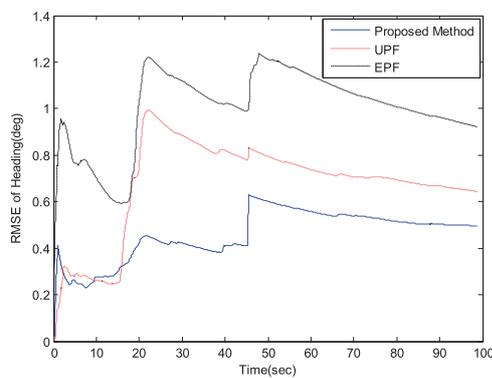


Figure 14. RMSE of heading of robot when the statistics of the noises are assumed to be unknown.

Experimental

The robot localization on the proposed algorithm is compared with localization based on EPF using the car park data set. The environment of this test is the top level of the car park building of the university campus. The full data set and the documentation is available at [33]. A standard utility vehicle is a four-wheeled vehicle equipped with Sick laser range and bearing sensor, linear variable differential transformer sensor for the steering, back wheel velocity encoder and GPS as shown in Figure 15.

In this experiment, artificial landmarks are used that consisted of 60 mm steel poles covered with reflective tape. The true position of the landmarks is also obtained with GPS and is known. In addition, GPS is used to provide accuracy ground truth that is used to evaluate the estimated path of the robot. When the vehicle was driven around the park, the velocity and steering angle is measured using encoders, but uneven terrain induced additional non-systematic errors because of wheel slippage and vehicle attitude. Hence, the trajectory of dead reckoning deviated from the ground truth provided by GPS as shown in Figure 16.

The trajectory is estimated by EPF and the proposed method is shown in Figure 17. As can be seen, the path errors of localization based on proposed method is lower than that of localization based on EPF,



Figure 15. Utility car used for the experiment.

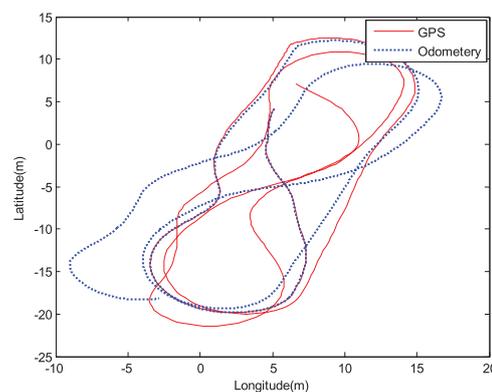


Figure 16. Odometry of the vehicle.

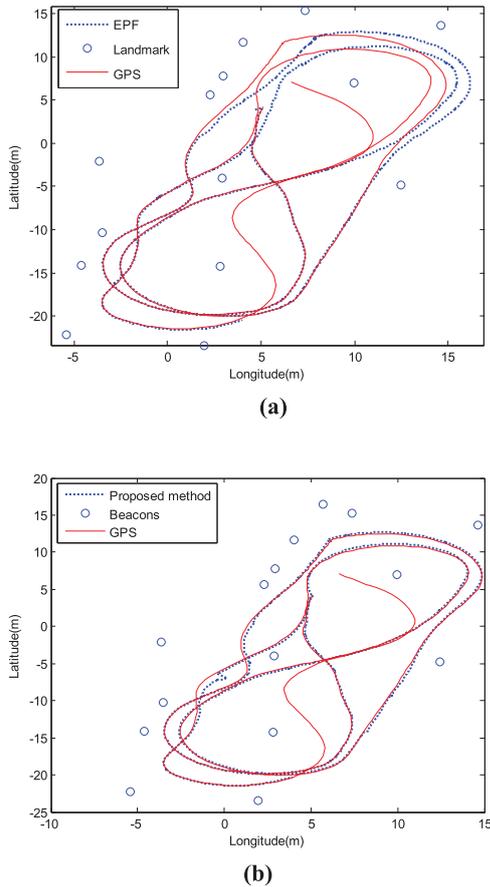


Figure 17. (a) EPF (b) Proposed Method.

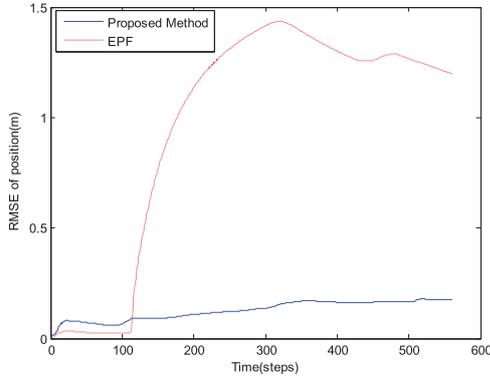


Figure 18. RMSE of position.

because the result of the localization based on proposed algorithm better matches with the GPS ground truth. Figure 18 shows the RMSE of the robot position. The results show that the performance of the proposed algorithm is better than that of EPF. The proposed algorithm starts with a wrongly known statistics noise and then adapts them through FIS and attempts to minimize the mismatch between the theoretical and actual values of the innovation sequence. Figures 14 and 15 show the comparison between the proposed algorithm and the EPF. This is because that the proposed method tunes Q_t and R_t adaptively and diversity particles are more than that of EPF. This improves data association, estimation accuracy and consistency.

Conclusion

The PF is a novel technique that has sufficiently good estimation results for the nonlinear/non-Gaussian systems. However, the general PF is inconsistent, what is caused mainly by unknown a priori knowledge of the noise statistics and loss of particle diversity in resampling step. The soft computing (PSO and FIS) may be used in order to overcome particle filtering weaknesses. In this paper, APF that takes advantage of soft computing in order to overcome the problems of PF is presented. The APF uses AFSRUKF to generate the proposal distribution, in which the covariance of the measurement and process of the state are adjusted online by predicted residual as an adaptive factor based on a covariance-matching technique using FIS. As a result, the proposed method can work in unknown statistical noise and does not require a priori knowledge about the system. In addition, to increase diversity, the resampling process is done based on PSO. The validity of the proposed method is evaluated against two distinct examples: the univariate growth model and robot localization. By simulation analysis and evaluation, it is verified that applying APF would contribute to significant estimation performance improvements.

Disclosure statement

No potential conflict of interest was reported by the author.

References

- [1] Chen S. Kalman filter for robot vision: A survey. *IEEE Trans. Ind. Electron.* 2012;59(11):4409–4420.
- [2] Dias SS, Bruno MG. Cooperative target tracking using decentralized particle filtering and RSS sensors. *IEEE Trans. Signal Proces.* 2013;61(14):3632–3646.
- [3] Hiremath SA, Van Der Heijden GW, Van Evert FK, et al. Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Comput. Electron. Agr.* 2014;100:41–50.
- [4] Atia M, Georgy J, Korenberg M, et al. Real-time implementation of mixture particle filter for 3D RISS/GPS integrated navigation solution. *Electron Lett.* 2010;46(15):1083–1084.
- [5] Arulampalam S, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* 2002;50(2):174–188.
- [6] Ristic B, Arulampalam S, Gordon N. *Beyond Kalman filter: particle filters tracking applications*. 1st ed. Boston: Artech House; 2004.
- [7] Loiola MB, Lopes RR, Romano JMT. Modified Kalman filters for channel estimation in orthogonal space-time coded systems. *IEEE Trans. Signal Proces.* 2012;60(1):533–538.
- [8] Roshany-Yamchi S, Cychowski M, Negenborn RR, et al. Kalman filter-based distributed predictive control of large-scale multi-rate systems: application to power networks. *IEEE Trans. Contr. Syst. T.* 2013;21(1):27–39.
- [9] Julier SJ, Uhlmann JK, Durrant-Whyte HF. A new method for the nonlinear transformation of means

- and covariances in filters and estimators. *IEEE Trans Automat Contr.* **2000**;45(3):477–482.
- [10] Park S, Hwang J, Kim E, et al. A new evolutionary particle filter for the prevention of sample impoverishment. *IEEE Trans. on Evolutionary Computation.* **2009**;13(4):801–809.
- [11] Rigatos GG. Particle and Kalman filtering for state estimation and control of DC motors. *ISA Trans.* **2009**;48:62–72.
- [12] Van Leeuwen PJ. Particle filtering in geophysical systems. *Mon. Weather Rev.* **2009**;137(12):4089–4114.
- [13] Merwe RVD, Doucet A, Freitas ND, et al. The unscented particle filter. Dept. Eng., Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 380; 2000.
- [14] Li DZ, Wang W, Ismail F. A mutated particle filter technique for system state estimation and battery life prediction. *IEEE Trans Instrum Meas.* **2014**;63(8):2034–2043.
- [15] Kotecha JH, Djuric P. Gaussian particle filtering. *IEEE Trans. Signal Process.* **2003**;51(10):2593–2602.
- [16] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat Comput.* **2000**;10(3):197–208.
- [17] Merwe R, Doucet A, de Freitas N, et al. The unscented particle filter. (Cambridge University Engineering Department), 2000, Technical Report CUED/F INFENG/TR 380.
- [18] Li W, Sun S, Yia J, et al. Robust unscented Kalman filter with adaptation of process and measurement noise covariances. *Digital Signal Processing.* **2016**;48:93–103.
- [19] Hajiyeve C, Soken HE. Robust adaptive unscented Kalman filter for attitude estimation of pico satellites. *Int. J. Adapt. Control Signal Process.* **2014**;28(2):107–120.
- [20] Mehra RK. On the identification of variances and adaptive Kalman filtering. *IEEE Trans Autom. Control.* **1970**;15(2):175–184.
- [21] Nam K, Oh S, Fujimoto H, et al. Estimation of sideslip and roll angles of electric vehicles using lateral tire force sensors through RLS and Kalman filter approaches. *IEEE Trans. Ind. Electron.* **2013**;60(3):988–1000.
- [22] Jiancheng F, Sheng Y. Study on innovation adaptive EKF for inflight alignment of airborne POS. *IEEE Trans. Instrum. Meas.* **2011**;60(4):1378–1388.
- [23] Tang XJ, Yan J, Zhong DD. Square-root sigma-point kalman filtering for spacecraft relative navigation. *Acta Astronautica.* **2010**;66(5):704–713.
- [24] Feng DW, Li T, Huang ZT. Square-root second-order extended Kalman filter and its application in target motion analysis. *Radar, Sonar & Navigation, IEF.* **2010**;4(3):329–335.
- [25] Merwe R, Wan E. The square-root unscented Kalman filter for state and parameter-estimation. *IEEE International Conference on Acoustics, Speech, and Signal Processing.* **2001**;6:3461–3464.
- [26] Fen C, Wang M, Ji QB. Analysis and comparison of resampling algorithms in particle filter. *J Syst Simul.* **2009**;21(4):1101–1105.
- [27] Bolic M, Djuric PM, Hong S. Resampling algorithms and architectures for distributed particle filters. *IEEE Trans. Signal Processing.* **2005**;53(7):2442–2450.
- [28] Fu X, Jia Y. An improvement on resampling algorithm of particle filters. *IEEE Trans. on Signal Processing.* **2010**;58(10):5414–5420.
- [29] Li T, Sattar TP, Sun S. Deterministic resampling: unbiased sampling to avoid sample impoverishment in particle filters. *Signal Processing.* **2012**;92(7):1637–1645.
- [30] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks.* **1995**;4:1942–1948.
- [31] Clerc M, Kennedy J. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Tran. Evolutionary Computation.* **2002**;6(1):58–73.
- [32] Havangi R, Taghirad HD, Nekoui MA. A square root unscented FastSLAM with improved proposal distribution and resampling. *IEEE Trans Ind Electron.* **2014**;61(5):2334–2345.
- [33] Nebot E. Victoria Park data set. Available from: <http://www-personal.acfr.usyd.edu.au/nebot/dataset.htm>.