TARGET TRACKING USING A PARTICLE FILTER BASED ON THE PROJECTION METHOD

Y. Zhai, M. Yeary and D. Zhou

School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK 73019 USA
Corresponding Email: {yan.zhai, yeary, dayong}@ou.edu

ABSTRACT

We present a new particle filter (PF) algorithm, which uses a mathematical tool known as Galerkin's projection method to generate the proposal distribution. By definition, Galerkin's method is a numerical approach to approximate the solution of a partial differential equation. By leveraging this method with L^2 theory and the FFT, this new proposal is fundamentally different to various local linearization or Kalman filter based proposals. We apply this algorithm to a bearings-only tracking problem. As shown in the theory and indicated by our simulations, this proposal renders more support from the true posterior distribution, thereby significantly enhances the estimation accuracy compared to standard bootstrap filters. In addition, because of this improved proposal distribution, the new particle filter can achieve a given level of performance with less sample size.

Keywords: Nonlinear filters, Tracking, Importance sampling

1. INTRODUCTION

Bearings-only target tracking is a fundamental component of surveillance, guidance or positioning systems, whose objective is to determine the target's kinematics, such as locations, velocities, etc., based on the angle-measurement history. Due to the inherent nonlinearities in the observation model, bearings-only target tracking has become a standard nonlinear filtering problem that receives intensive investigations. Traditionally, various Kalman filter based tracking techniques have been used. However, for many cases these methods cannot provide satisfactory results. More recently, particle filtering techniques have received increasing attention [1][2][3]. Bearing the nature of sequential importance sampling and the Monte Carlo approach, particle filtering has emerged as a superior alternative to the traditional tracking methods. The basic idea of the particle filter is to approximate the pdf of the system state by a set of weighted samples (called particles) generated from a proposal distribution. Within a sequential framework, the particles and their weights are propagated through the system and updated whenever the most recent measurements are received. A particle filter involves sequential importance sampling (SIS) and resampling steps. However, the performances of these techniques are strongly influenced by the choice of the proposal distribution.

In this paper, we propose a new particle filter algorithm in which the Galerkin's projection method is used to generate the proposal distribution. Applying the Galerkin's method to the nonlinear filtering problem has been reported in [4] [5]. The rationale behind Galerkin's method is to assume the state posterior distribution is in L^2 space. In this case, this distribution can be approximated by its projection onto a finite set of orthogonal basis. At each iteration, it only needs to update the projection on each basis to approximate the true proposal distribution. In addition, by choosing a set of special exponential basis, the projection can be approximated by the FFT which is computationally efficient to implement. Finally, we use this approximated distribution as the proposal in our new PF algorithm. This proposal does not require any local linearization of the nonlinear system and does not have any Gaussian assumption of the systems' states when calculating the proposal, which differs fundamentally to various Kalman filter based PF algorithms in [6] [7] [8]. Because of this improved proposal distribution, the new particle filter can yield accurate estimations while using less particles.

2. PROPOSED PARTICLE FILTERING BASED ON GALERKIN'S METHOD

In this section, we'll present how Galerkin's method [4] can be used to generate the proposal distribution, and how it can be incorporated into the particle filter framework. For the sake of completeness and to facilitate our derivation, we briefly review the setup of nonlinear filtering and particle filters in this section. Consider a nonlinear system given as follows:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{v}_{t-1}, \quad (1)$$
$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{n}_t, \quad (2)$$

where $\mathbf{x}_t$ and $\mathbf{y}_t$ denote the hidden states and the observations of the system at time t, respectively. Both $f(\cdot)$ and $h(\cdot)$ could be nonlinear functions, and $\mathbf{v}_t$ and $\mathbf{n}_t$ denote the process and observation noises, respectively. The objective is to estimate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ governed by Chapman-Kolmogorov equation and Bayes' rule given below:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \ , \quad (3)$$
$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})dx_t} \ . \quad (4)$$

In a PF framework, the posterior distribution can be approximated by a set of weighted points:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N_s} \tilde{\omega}_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \ ,$$

Partial funding for this research was received from the Office of Naval Research under DEPSCoR grant N00014-06-1-0590 and from the National Oceanic & Atmospheric Administration under cooperative agreement number NA17RJ1227.

where $\tilde{\omega}_t^{(i)} = \omega_t^{(i)} / \sum_{j=1}^{N_s} \omega_t^{(j)}$ is the normalized importance weight, and $\omega_t^{(i)}$ is given as:

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})} \quad . \tag{5}$$

Equation (5) is the particle weight update equation. Distributions $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ and $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})$ represent the system's likelihood and the state transition prior, respectively. The particles (or the samples) are drawn from a proposal distribution as $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{0:t})$. In a standard PF algorithm, *i.e.* the bootstrap filter, the transition prior $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})$ is used as the proposal distribution. This algorithm is easy to implement, but its proposal does not include the most recent measurement information, which may cause the filter to diverge from the true target trajectory, especially when the system is highly nonlinear [1]. Designing a proposal is a challenging task of current research, and it's also the topic of this paper. A good proposal needs to have a strong support from the true posterior distribution, and should be easy to sample from. In the following subsections, *we'll show that Galerkin method can be used to generate the better proposal for particle filters*.

2.1. Generating Proposal Using Galerkin's Method

Galerkin's method is a numerical approach to approximate the solution of a partial differential equation (PDE) [4][5]. Let $\mathcal{P}(x,t) = 0$ denotes a PDE, which is a function of both temporal variable t and spatial variable x. The basic idea of Galerkin's method is to assume that $p(x,t)$ is the *solution* of the above PDE, and it is in the L^2 space, such that it can be decomposed by the following equation:

$$p(x,t) = \sum_{l=0}^{\infty} \epsilon_l(t)\phi_l(x) \quad , \tag{6}$$

where $\{\phi_l(x)\}_{l=0}^{\infty}$ is a set of complete orthogonal basis of the L^2 space and $\epsilon_l(t)$ is the projection of $p(x,t)$ onto basis $\phi_l(x)$ at time t defined by the inner product $\langle p(x,t), \phi_l(x)\rangle$ as:

$$\langle p(x,t), \phi_l(x)\rangle = \int p(x,t)\phi_l(x)^* dx \quad . \tag{7}$$

Our objective is to find an approximation of $p(x,t)$, denoted as $\hat{p}(x,t)$, such that

$$\hat{p}(x,t) = \sum_{l=0}^{N-1} c_l(t)\phi_l(x) \quad . \tag{8}$$

Note that the approximation error arises from the replacement of infinite basis with a set of finite orthogonal basis. The projections $c_l(t), l = 0, \cdots, N-1$, are the values to be determined. Having this setup, we can project $\mathcal{P}(x,t)$ onto the subspace $\mathrm{span}\{\phi_l(x)\}_{l=0}^{N-1}$ as:

$$\langle \mathcal{P}(x,t),\ \phi_l(x)\rangle = 0, \qquad l = 0, \cdots, N-1 \quad . \tag{9}$$

By doing this, we convert solving the PDE into solving N ordinary differential equations (ODE). Next we apply this method to the nonlinear filtering problem defined previously.

First we assume $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{l=0}^{N-1} \tilde{c}_l(t)\phi_l$, where $\tilde{c}_l(t)$ will be determined later. For simplification of notations, we drop the variable x. Then we apply Galerkin's method to equation (4) by projecting it onto $\mathrm{span}\{\phi_l(x)\}_{l=0}^{N-1}$ as:

$$\begin{aligned}\langle p(\mathbf{x}_t|\mathbf{y}_{1:t}), \phi_k\rangle &= \sum_{l=0}^{N-1} c_l(t)\langle \phi_l, \phi_k\rangle \\ &= \frac{\sum_{l=0}^{N-1} \tilde{c}_l(t)\langle p(\mathbf{y}_t|\mathbf{x}_t)\phi_l, \phi_k\rangle}{\sum_{l=0}^{N-1} \tilde{c}_l(t)\langle p(\mathbf{y}_t|\mathbf{x}_t), \phi_l^*\rangle}\end{aligned} \tag{10}$$

where $k = 0, \cdots, N-1$. For simplification, the above equation can be written in a matrix form as:

$$\mathbf{C}(t) = \frac{\boldsymbol{\Upsilon}_t \tilde{\mathbf{C}}(t)}{\upsilon_t^T \tilde{\mathbf{C}}(t)} \tag{11}$$

where $\boldsymbol{\Upsilon}_t$ is a $N \times N$ matrix with the element at k^{th} row and l^{th} column given by $[\boldsymbol{\Upsilon}_t]_{k,l} = \langle p(\mathbf{y}_t|\mathbf{x}_t)\phi_l, \phi_k\rangle$. The variables $\mathbf{C}(t)$, $\tilde{\mathbf{C}}(t)$ and υ_t are $N \times 1$ vectors, with $[\upsilon_t]_l = \langle p(\mathbf{y}_t|\mathbf{x}_t), \phi_l^*\rangle$. Now we choose the exponential basis as $\phi_l(x) = \frac{1}{\sqrt{b-a}} \exp\left(j2\pi l \frac{x-a}{b-a}\right)$, where a and b are the integral limits. It has been shown in [4] that by using this set of basis the inner product can be approximated by FFT as follows:

$$\begin{bmatrix} \langle p(x), \phi_0\rangle \\ \vdots \\ \langle p(x), \phi_{N-1}\rangle \end{bmatrix} \approx \frac{\sqrt{b-a}}{N} \mathrm{FFT}[p(x)]$$

$$\begin{bmatrix} \langle p(x), \phi_0^*\rangle \\ \vdots \\ \langle p(x), \phi_{N-1}^*\rangle \end{bmatrix} \approx \sqrt{b-a}\ \mathrm{IFFT}[p(x)] \quad ,$$

see [4] [5] for details. Then, equation (11) can be approximated by using FFT as follows:

$$[\boldsymbol{\Upsilon}_t]_l = (\sqrt{b-a}/N)\ \mathrm{FFT}\ [p(\mathbf{y}_t|\mathbf{x}_t)\ \phi_l] \tag{12}$$

$$\upsilon_t = \sqrt{b-a}\ \mathrm{IFFT}\ [p(\mathbf{y}_t|\mathbf{x}_t)] \quad . \tag{13}$$

To evaluate $\tilde{c}_l(t)$, we apply the Galerkin's method in a similar way to equation (3). Then we have:

$$\tilde{c}_l(t) \approx (\sqrt{b-a}/N)\mathrm{IFFT}_l\left[c_l(t-1)\mathrm{FFT}_l[p(\mathbf{x}_t|\mathbf{x}_{t-1})]\right] , \tag{14}$$

where the $\mathrm{FFT}_l[\cdot]$ represents the l^{th} bin of FFT of the argument. In addition, $\tilde{c}_l(t)$ can also be calculated by:

$$\tilde{\mathbf{C}}(t) = \mathrm{FFT}\left[\sqrt{b-a} \cdot p(\mathbf{x}_t|\mathbf{x}_{t-1})\ \mathrm{IFFT}[\mathbf{C}(t-1)]\right] \quad . \tag{15}$$

Moreover, the prediction distribution and posterior distribution can be calculated by:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\tilde{\mathbf{C}}(t)] \tag{16}$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\mathbf{C}(t)] \quad . \tag{17}$$

As a summary, in order to approximate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, we only need to update the vector $\mathbf{C}(t)$ at each iteration.

2.2. Proposed New PF Algorithm

In this section, we incorporate Galerkin's method within the particle filter framework. More specifically, at each iteration, we use $\tilde{\mathbf{C}}(t)$ and $\mathbf{C}(t)$ to approximate the posterior distribution leveraging the IFFT defined in (17). Then draw particles from this approximated distribution. After that, the particles' weights will be evaluated. The final step is the resampling and update stage. Since the proposal is generated by projecting the true posterior distribution onto a subspace of L^2 space, the accuracy of the proposal is guaranteed by choosing appropriate the number of basis. As indicated in our simulation, using a limited number of basis is good enough to generate accurate proposal which in turn significantly reduces the number particles used to achieve a given level of performance. The detailed algorithm is summarized in as follows:

The Galerkin Method Based PF algorithm:

- Sequential Importance Sampling (SIS) Step:
  - Calculate the parameters $\tilde{\mathbf{C}}(t)$ and $\mathbf{C}(t)$ using (12) to (14) or (15);
  - Generate the proposal distribution using (17);
  - Sample from the proposal distribution according to
  $$\begin{aligned} x_t^{(i)} &\sim q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}) \\ &\approx p(\mathbf{x}_t|\mathbf{y}_{1:t}) \\ &\approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\mathbf{C}(t)]; \end{aligned}$$
  - Evaluate and normalize the importance weights according to (5);
- Resampling Step: Generate a new set of particles $x_t^{i\star}$ from $x_t^{(i)}$ by sampling N_s times the approximate distribution of so that $Pr\left(x_t^{i\star} = x_t^{(j)}\right) = \tilde{\omega}_t^{(j)}$;
- Output and Update Step: Approximate x_t by $\hat{x}_t \approx \frac{1}{N_s}\sum_{i=1}^{N_s} x^{i\star}(t)$ and update the proposal.

3. SIMULATIONS RESULTS

In this section, we apply the proposed particle filter to a single sensor bearings-only tracking problem. Bearings-only tracking has many practical applications, such as passive sonar applications and aircraft surveillance. Moreover, this is a standard nonlinear problem that has been intensively investigated in current literature [2][3]. The system state of the target is represented by a vector given as:

$$\mathbf{X}_t = \left[\begin{array}{cccc} x(t) & y(t) & v_x(t) & v_y(t) \end{array}\right]^T . \quad (18)$$

The variables (x_t, y_t) are the target location in a Cartesian coordinates. The variables $v_x(t)$ and $v_y(t)$ denote the target velocities along the x-axis and y-axis, respectively. The discrete time model for the kinematics of a non-maneuvering target is given as [3]:

$$\mathbf{X}_t = \mathbf{F}\cdot\mathbf{X}_{t-1} + \Gamma\cdot\mathbf{V}_{t-1} \ , \quad (19)$$

where

$$\mathbf{F} = \left[\begin{array}{cccc} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right] \quad \Gamma = \left[\begin{array}{cc} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{array}\right] ,$$

and T is the sampling rate. The process noise $\mathbf{V}_t$ is a 2-by-1 noise vector with a distribution $\mathbf{V}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_v)$, where

$$\mathbf{Q}_v = \left[\begin{array}{cc} \sigma_v^2 & 0 \\ 0 & \sigma_v^2 \end{array}\right] .$$

The measurement of the tracking system is the angle between the observation platform (the coordinate origin) and the location of the target, which is given as:

$$z(k) = \arctan\left[\frac{x(t)}{y(t)}\right] + n(t) \quad (20)$$

where the scalar variable $n(t)$ denotes the measurement noise which is distributed as $n \sim \mathcal{N}(0, \sigma_n^2)$. Equations (19) and (20) constitute the state space model for the bearings-only tracking problem, in which the observation model contains nonlinearities. In addition, the velocities $v_x(t)$ and $v_y(t)$ are the hidden states of the system, which do not have direct measurements.

In this section, we provide two tracking examples to test the performance of the proposed particle filter tracking algorithm. In the first example, the target has a approximate rectilinear motion. While in the second example, the target makes a maneuver operation. In addition, for the purpose of comparison, an extended Kalman filter (EKF) and a bootstrap filter (a PF algorithm using state transition prior as proposal) are also implemented. In the first example, the new PF algorithm uses 100 basis and 200 particles, i.e. $N = 100, N_s = 200$. While the bootstrap filter uses the sample size $N_s = 200, 500, 1000, 2000$, respectively. To evaluate the tracking performance of these filters, 200 Monto Carlo runs are implemented. In addition, the root mean square error (RMSE) along

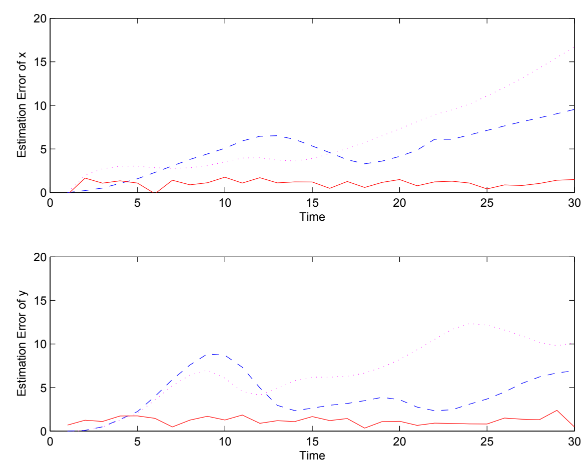

Fig. 1. *The estimation RMSE vs. time for both x-axis and y-axis: the solid line(—) denotes the RMSE of the proposed PF; the dashed line (– –) denotes the RMSE of the bootstrap filter with 200 particles; the dotted line (. . .) represents the RMSE of the EKF.*

both x-axis and y-axis are used as performance indices. The tracking results of the three filters are shown in Figure 1, Figure 2 and Table 1. In Figure 1, the RMSE's vs. time from the 200 Monte Carlo run are plotted. It is obvious from this figure, the new PF gives much more accurate estimation compared to the bootstrap filter using the same sample size. In addition, the tracking results from a typical realization is shown in Figure 2. As shown in this figure, although the bootstrap filter can following the general direction of the target, it fails to detect small maneuvering operations made by the target. On the other hand, due to the improved proposal distribution, the new proposal PF can keep a close track throughout the whole simulation. Next, for a comparison, bootstrap filters with different sample size are implemented 200 times to generate the performance index. Table 1. summarizes the mean of and the variance of RMSE's for each bootstrap filter. It is observed from this table that even a bootstrap with a sample size of 2000 still cannot achieve the estimation accuracy of the new proposed PF algorithm.

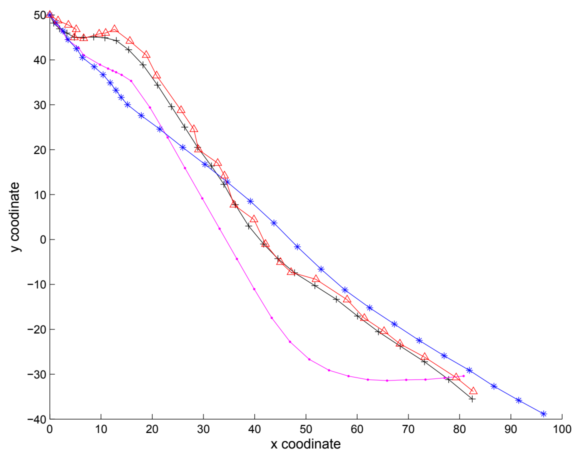

Fig. 2. *The first example: the line with a plus sign (—+—) represents the true trajectory; the line with a triangle (–△–) represents the estimated trajectory from the new PF; (–●–) represents the tracking results from EKF; (–∗–) represents the tracking results from bootstrap filter. As seen, the our proposed PF method has the closed track.*

Table 1. The comparison of the new PF with the Bootstrap filter

Filter (Sample Size)	Mean RMSE		Var RMSE	
	e_x	e_y	Var e_x	Var e_y
Bootstrap (200)	4.8312	4.0215	6.7798	5.7492
Bootstrap (5000)	4.0599	3.8300	4.0483	6.0063
Bootstrap (1000)	3.6181	3.6961	3.3728	5.8839
Bootstrap (2000)	3.6199	3.6489	3.3991	6.0310
New PF (200)	1.1391	1.1554	0.0535	0.0576

In the second example, we use the three filters to track a target which makes a sharp turn approximately at the location (x=72,y=6). This kind of target is always referred as a maneuvering target. Tracking a maneuvering target is more challenging than tracking a target with constant velocities. Traditionally, multiple model methods are used in the case. However, to test the accuracy and robustness of our new PF, we still use the single constant velocity model given before. The tracking result of one typical realization is shown in Figure 3. As indicated in this figure, both the EKF and the bootstrap filter diverges from the true trajectory when the target makes the sharp turn. However, the new PF still can keep a close track.

4. CONCLUSION

In this paper, we proposal a new PF algorithm which uses Galerkin's projection method to generate a proposal distribution. This method

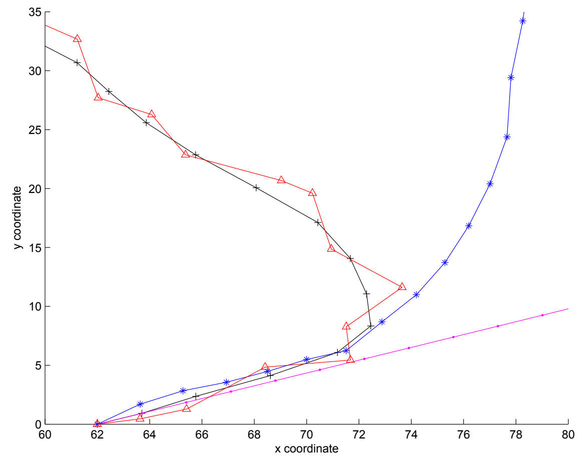

Fig. 3. *The second example: The line with a plus sign (+) represents the true trajectory; the line with a triangle (–△–) represents the estimated trajectory from the new PF; (–●–) represents the tracking results from EKF; (–∗–) represents the tracking results from bootstrap filter with 500 particles. As seen, the our proposed PF method has the closest track.*

renders a proposal which has more support from true posterior distribution. This algorithm is implemented to two tracking examples. As indicated by the simulation results, this filter outperforms the standard bootstrap filter, can achieve a given level of performance with fewer samples.

5. REFERENCES

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

[2] A. Doucet, N. de Freitas, and N. Gordon, "Sequential Monte Carlo Methods in Practice," Springer, New York, 2001.

[3] B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, Feb. 1, 2004.

[4] J. Gunther, R. Beard, J. Wilson, T. Oliphant and W. Stirling, "Fast nonlinear filtering via Galerkin's method," *American Control Conference (ACC)*, pp. 2815-2819, 1997.

[5] R. Beard, J. Gunther, J. Lawton and W. Stirling, "Nonlinear projection filter based on Galerkin approximation," *AIAA J. of Guidance, Control and Dynamics*, vol: 22, no.2, pp. 258-266, 1999.

[6] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo Sampling methods for Bayesian filtering," *Statist. and Comput.,* 10(3), pp. 197-208, 2000.

[7] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," *IEEE Transactions on Signal Processing*, pp. 613624, 2001.

[8] R. van der Merwe, A. Doucet, N. de Freitas and E. Wan, "The unscented particle filter," *Technical Report CUED/F-INFENG/TR 380,* Dept. of Eng., Cambridge University, 2000.