

بررسی مسئله تخمین از دیدگاه بهینه‌سازی مقید و طراحی تخمین‌گر تکاملی

رمضان هاونگی^۱، محمد تشنه لب^۲، محمد علی نکویی^۳ و حمید رضا تقی راد^۴

دانشکده مهندسی برق و کامپیوتر

دانشگاه صنعتی خواجه نصیرالدین طوسی

(تاریخ دریافت: ۸۸/۰۹/۳۰؛ تاریخ پذیرش: ۹۰/۰۲/۱۱)

چکیده

در این مقاله، به مسئله تخمین به صورت یک مسئله بهینه‌سازی مقید توجه شده و سپس با استفاده از روش‌های تکامل تفاضلی (DE) و اجتماع ذرات (PSO) حل شده است. فیلتر پیشنهادی به طور تصادفی در فضای حالت‌های سیستم شروع به جستجو می‌کند و بهترین تخمین را در هر لحظه به دست می‌آورد. عملکرد فیلتر پیشنهادی مبتنی بر تکامل تفاضلی و اجتماع ذرات با فیلتر کالمن توسعه یافته (EKF) و فیلتر ذره‌ای، برای یک سیستم غیرخطی در شرایط مختلف با هم مقایسه شده است. نتایج شبیه‌سازی‌ها نشان می‌دهد که عملکرد فیلترهای تکاملی در شرایط مختلف به مراتب بهتر از فیلتر کالمن توسعه یافته و فیلتر ذره‌ای است.

واژه‌های کلیدی: فیلتر کالمن توسعه یافته، فیلتر بیز، ماکزیمم کردن تابع چگالی احتمال پسین، تکامل تفاضلی، اجتماع ذرات

A Study of Estimation Problem from Viewpoint of Conditional Optimization and Designing of Evolutionary Estimator

R. Havangi, M. Teshnehlab, M.A. Nekoui, and H. Taghirad

Elec. & Comp. Eng. Dep't.

K.N. Toosi Univ. of Tech.

(Received: Dec. 21, 2009; Accepted: May 1, 2011)

ABSTRACT

In this paper, estimation problem is considered as a conditional optimization problem and then is solved using both differential evolution (DE) and particle swarm optimization (PSO). The proposed filter searches stochastically along the state space and obtains the best estimation at any time. The performance of the proposed filter, based on differential evolution (DE) and particle swarm optimization (PSO), is compared with extended Kalman filter (EKF) and with particle filter in different conditions for a non-linear system. Simulation results show that the performance of evolution filters in different conditions is superior to extended Kalman and particle filters.

Keywords: Extended Kalman Filter, Bayes Filter, Maximum A Posteriori, Differential Evolution, Particle Swarm Optimization

۱- دانشجوی دکتری: Havangi@ee.kntu.ac.ir

۲- دانشیار (نویسنده پاسخگو): Teshnehlab@eed.kntu.ac.ir

۳- استادیار: Manekoui@eed.kntu.ac.ir

۴- دانشیار: Taghirad@kntu.ac.ir

۱- مقدمه

وجود مزایایی که دارد، دارای نقاط ضعف بزرگی است، که از جمله این نقاط ضعف می‌توان به مسئله فقر نمونه [۱۲]، تباہیدگی [۵]، انتخاب تابع توزیع پیشنهادی [۱۳ و ۶] و وابستگی به تعداد ذرات [۲-۳] اشاره کرد. به منظور حل این مشکلات، در این مقاله یک روش جدید پیاده‌سازی عملی فیلتر بیز برای تخمین حالت‌های سیستم‌های غیر خطی بر اساس الگوریتم‌های تکامل تفاضلی^۸ (DE) و اجتماع ذرات^۹ (PSO) ارائه شده است. در روش پیشنهادی، به مسئله تخمین به صورت یک مسئله بهینه‌سازی مقید نگریسته شده و سپس برای حل این مسئله از الگوریتم تکامل تفاضلی و اجتماع ذرات استفاده شده است. الگوریتم پیشنهادی به طور تصادفی در فضای حالت شروع به جستجو می‌کند و بهترین تخمین حالت‌ها را در هر لحظه به دست می‌آورد.

۲- مسئله تخمین

سیستم غیر خطی زیر را در نظر بگیرید:

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (1)$$

$$z_t = g_t(x_t) + v_t, \quad (2)$$

که در آن، x_t ، u_t و z_t به ترتیب متغیرهای حالت، ورودی کنترلی و اندازه‌گیری سیستم در لحظه t می‌باشند. همچنین فرض می‌شود که نویز پروسه w_t و اندازه‌گیری v_t مستقل از یکدیگر با توزیع گوسی باشند. مسئله تخمین، شامل پیدا کردن بهترین تخمین متغیرهای حالت x_t بر اساس همه اندازه‌گیری‌های قابل دسترس $Y_t = \{z_{1:t}, u_{1:t}\}$ است. همان طور که گفته شد از دیدگاه تئوری بیزین مسئله تخمین عبارت از تخمین تابع چگالی احتمال پسین است. روش بیزین تابع چگالی احتمال پسین را به صورت بازگشتی طی دو گام به صورت زیر تخمین می‌زند:

۱- مرحله به‌روزرسانی^{۱۰}

این مرحله توسط قانون بیز به صورت زیر انجام می‌شود:

$$p(x_t | Y_t) = \frac{p(z_t | x_t, Y_{t-1})p(x_t | Y_{t-1})}{p(z_t | Y_{t-1})} \quad (3)$$

$$= \frac{p(z_t | x_t)p(x_t | Y_{t-1})}{p(z_t | Y_{t-1})}$$

مسئله تخمین سیستم‌های غیر خطی، موضوع مهمی در بسیاری از کاربردها می‌باشد. فیلتر بیز^۱ مرسوم‌ترین روش تخمین است. از دیدگاه تئوری بیزین مسئله تخمین عبارت از تخمین تابع چگالی احتمال پسین^۲ است. با دانستن چگالی احتمال پسین می‌توان تخمین بهینه حالت‌ها را نسبت به هر تابع معیاری محاسبه نمود. اما این فیلتر دارای یک حل بسته نیست و از این رو جهت پیاده‌سازی عملی مناسب نمی‌باشد. بسته به مدل فرآیند و اندازه‌گیری، روش‌های مختلفی برای حل عملی فیلتر بیز وجود دارد. به عنوان مثال اگر مدل سیستم دینامیکی مورد بررسی، به صورت خطی باشد و نویزهای فرآیند و اندازه‌گیری دارای ماهیت گوسی باشند از فیلتر کالمن استفاده می‌شود [۱]. برای سیستم‌های دینامیکی خطی محدود از فیلترهای مبتنی بر جدول^۳ استفاده می‌شود [۲-۳]. در صورتی که سیستم غیرخطی و دارای نویز سفید گوسی باشد، می‌توان از فیلتر کالمن توسعه‌یافته استفاده کرد [۴-۵]. در صورتی که میزان غیر خطی سیستم زیاد باشد فیلتر کالمن توسعه‌یافته به صورت بهینه عمل نخواهد کرد و تخمین‌ها بدی خواهند داشت. غیر خطی بودن بالای سیستم سبب می‌شود که تقریب خطی سیستم غیرخطی بد شده و در نتیجه تابع توزیع احتمال پسین از حالت گوسی بودن بسیار منحرف و سبب خراب شدن تخمین‌ها شود [۴-۶]. دیگر راه حل عملی برای پیاده‌سازی فیلتر بیز، روش‌های غیر پارامتریک می‌باشد. فیلتر ذره‌ای، مهم‌ترین روش غیرپارامتریک برای تخمین سیستم‌های غیرخطی است که محققان زیادی بر روی آن کار نموده‌اند [۷-۱۱]. فیلتر ذره‌ای، در اصل پیاده‌سازی مبتنی بر نمونه برداری^۴ است که در آن تابع چگالی احتمال پسین با مجموعه‌ای از ذرات وزن داده شده تخمین زده می‌شود [۴-۶]. فیلتر ذره‌ای کاربرد وسیعی در تخمین سیستم‌های غیرخطی و غیرگوسی دارد، که از آن جمله می‌توان کاربرد فیلتر ذره‌ای در موقعیت‌یابی^۵، نقشه-یابی^۶، ردیابی و تشخیص خطا^۷ را نام برد. فیلتر ذره‌ای با

1- Bayesian Field

2- Probability Density Function Posterior

3- Grid-Based Filter

4- Sample Based

5- Localization

6- Mapping

7- Fault Diagnosis

8- Differential Evolution

9- Particle Swarm Optimization

10- Update

پایه‌سازی برای فیلتر بیز است. در فیلتر ذره‌ای تابع چگالی احتمال پسین $p(x_t | Y_t)$ به صورت مجموعه‌ای از ذرات وزن داده شده (که تخمین حالت‌ها از ذرات و وزنشان به دست می‌آید) به صورت زیر بیان می‌شود:

$$S_t = \{(x_t^m, w_t^m) | m = 1, \dots, N\}. \quad (۹)$$

در این حالت $p(x_t | Y_t)$ با مجموع وزن داده شده‌ای به صورت زیر تقریب زده شود:

$$p(x_t | Y_t) \approx \sum_{i=1}^n w_t^{(i)} \delta(x_t - x_t^{(i)}), \quad (۱۰)$$

که $\delta(x)$ تابع دلتای دیراک، $w_t^{(i)} > 0$ وزن مربوط به $x_t^{(i)}$ و $\sum_{i=1}^n w_t^{(i)} = 1$ است. با وجود مزایایی که فیلتر ذره‌ای در

تخمین سیستم‌های غیر دارد، دارای نقاط ضعف بزرگی است. یکی از این نقاط ضعف این است که حتی با تعداد زیادی ذرات ممکن است هیچ ذره‌ای در نزدیک حالت صحیح قرار نگیرد. این ضعف در متون مربوط به فیلتر ذره‌ای به مسئله تباهیدگی^۶ معروف است. از طرف دیگر به دلیل اینکه پروسه نمونه برداری در الگوریتم فیلتر ذره‌ای بهینه نیست، دو مشکل در فیلتر ذره‌ای به وجود می‌آید. اولین مشکل مسئله فقر نمونه^۷ است که وقتی تابع درست نمایی^۸ خیلی باریک باشد و یا در دم توزیع پیشین^۹ قرار گیرد، اتفاق می‌افتد. این مشکل سبب می‌شود که بعد از مرحله به‌روزرسانی بیشتر ذرات وزن قابل توجه‌ای نداشته باشند. دیگر مشکل فیلتر ذره‌ای وابستگی به تعداد ذرات است. فیلتر ذره‌ای شدیداً به تعداد ذراتی که برای تخمین توزیع احتمال به کار می‌روند وابسته است و دقت تخمین به تعداد ذرات بستگی دارد.

محققان زیادی برای حل این مشکلات کار کرده‌اند. در مرجع [۱۵] از UKF برای بهبود تابع توزیع پیشنهادی استفاده شده است. در مراجع [۱۳ و ۱۸-۱۶] از الگوریتم ژنتیک برای بهبود عملکرد فیلتر ذره‌ای استفاده شده است. مرجع [۱۹] هم تلاش‌هایی برای بهبود عملکرد فیلتر ذره‌ای انجام داده است. در این مقاله به منظور رهایی از این مشکلات به مسئله تخمین به صورت یک مسئله بهینه‌سازی مقید نگریسته

در اینجا، $p(z_t | Y_{t-1})$ وابسته به تابع درست نمایی^۱ $p(z_t | x_t)$ مطابق رابطه زیر است:

$$p(z_t | Y_{t-1}) = \int p(z_t | x_t) p(x_t | Y_{t-1}) dx_t, \quad (۴)$$

که تابع درست نمایی $p(z_t | x_t)$ با توجه به معادله اندازه‌گیری و نویز اندازه‌گیری v_t قابل محاسبه است. ۲- مرحله پیش بینی^۲

این مرحله همان معادله معروف چاپمن-گولموگروف^۳ و به صورت زیر است:

$$p(x_{t+1} | Y_t) = \int p(x_{t+1} | x_t, u_t, Y_t) p(x_t | Y_{t-1}) dx_t, \\ = \int p(x_{t+1} | x_t, u_t) p(x_t | Y_{t-1}) dx_t. \quad (۵)$$

با دانستن چگالی احتمال پسین می‌توان تخمین بهینه حالت‌ها را نسبت هر تابع معیاری محاسبه نمود. به عنوان مثال می‌توان تخمین بهینه را با مینیمم کردن میانگین مربع خطا^۴ (MMSE) به صورت زیر به دست آورد:

$$\hat{x}_{MS} = \arg \min_{x_t^*} \int (x_t - x_t^*) (x_t - x_t^*)^T p(x_t | Y_t) dx_t, \quad (۶)$$

که ثابت می‌شود جواب معادله (۶) به صورت زیر است [۱۴]:

$$\hat{x}_{MS} = E[x_t | Y_t] = \int x_t p(x_t | Y_t) dx_t, \quad (۷)$$

و یا می‌توان تخمین بهینه را با ماکزیمم کردن تابع احتمال چگالی پسین (MAP) به دست آورد [۱۴]:

$$\hat{x}^{MAP} = \arg \max_{x_t} p(x_t | Y_t). \quad (۸)$$

با توجه به معادلات فیلتر بیز ملاحظه می‌شود که این فیلتر یک روش ادراکی است و از آنجایی یک حل تحلیلی و بسته برای آن وجود ندارد، از این رو مناسب پیاده‌سازی نمی‌باشد. در صورتی که معادلات سیستم (۱) و اندازه‌گیری (۲) خطی بوده و نویز پروسه و اندازه‌گیری گوسی با میانگین صفر باشند، فیلتر کالمن بهترین روش پارامتریک برای حل فیلتر بیز می‌باشد [۴]. در صورتی که سیستم غیر خطی باشد از روش‌های زیر بهینه و تقریبی برای تخمین استفاده می‌شود که مهم‌ترین آن‌ها فیلتر کالمن توسعه‌یافته^۵ (EKF) [۵] و فیلتر ذره‌ای [۴ و ۶] است. فیلتر ذره‌ای یک روش عددی قابل

6- Degeneracy

7- Sample Impoverishment

8- Likelihood

9- Priori

1- likelihood

2- Prediction

3- Chapman-Kolmogorov

4- Minimizing Mean Square Error

5- Extended Kalman Filter

که از دیدگاه برنامه‌ریزی خطی و غیرخطی مطابق فرم استاندارد یک مسئله بهینه‌سازی مقید است. در صورتی که $f_0(x)$ بیان‌کننده تابع هدف باشد، داریم:

$$f_0(x) = \prod_{i=1}^t p(z_i | x_i) \prod_{i=1}^t p(x_i | x_{i-1}, u_{i-1}) p(x_0). \quad (17)$$

بدون از دست دادن کلیات مسئله در صورتی که معادلات سیستم و اندازه‌گیری به فرم معادلات (۱) و (۲) باشند در این حالت معادله $f_0(x)$ به صورت زیر خواهد شد:

$$f_0(x) = \prod_{i=1}^t p_v(z_i | x_i) \prod_{i=1}^t p_w(x_i | x_{i-1}, u_{i-1}) p(x_0). \quad (18)$$

در رابطه بالا p_v و p_w به ترتیب بیان‌کننده تابع چگالی احتمال برای نویز اندازه‌گیری و پروسه می‌باشند. در این صورت مسئله تخمین عبارت از پیدا کردن x در فضای شدنی مسئله به گونه‌ای که تابع $f(x)$ را در میان همه x ها ماکزیمم کند. جواب x باید به گونه‌ای باشد که ضمن ماکزیمم کردن $f(x)$ قیود $x_{t+1} = f(x_t, u_t) + w_t$ و $z_t = h(x_t) + v_t$ را برآورده سازد. با لگاریتم گرفتن از طرفین معادله (۱۴)، به دست می‌آید:

$$f(x_t) = \log f_0(x) = \sum_{i=1}^t \log p_v(z_i | x_i) + \quad (19)$$

$$\sum_{i=1}^t \log p_w(x_i | x_{i-1}, u_{i-1}) + \log p(x_0) \\ = \log p_v(z_t | x_t) + \log p_w(x_t | x_{t-1}, u_{t-1}) + \\ \sum_{i=1}^{t-1} \log p_v(z_i | x_i) + \sum_{i=1}^{t-1} \log p_w(x_i | x_{i-1}, u_{i-1}) + \log p(x_0).$$

از طرفی، $f(x_{t-1})$ را می‌توان به صورت زیر نوشت:

$$f(x_{t-1}) = \sum_{i=1}^{t-1} \log p_v(z_i | x_i) + \quad (20)$$

$$\sum_{i=1}^{t-1} \log p_w(x_i | x_{i-1}, u_{i-1}) + \log p(x_0).$$

بنابراین:

$$(21)$$

$f(x_t) = \log p_v(z_t | x_t) + \log p_w(x_t | x_{t-1}, u_{t-1}) + f(x_{t-1})$.
همچنین مسئله بهینه‌سازی مورد نظر یک مسئله بازگشتی است و با توجه به اصل بهینگی بلمن مسئله MAP را می‌توان به صورت زیر دوباره فرمول‌بندی کرد:

شده و سپس از الگوریتم‌های تکاملی برای حل این مسئله استفاده شده است. اساس روش پیشنهادی بر مبنای ماکزیمم کردن تابع احتمال پسین (MAP) با استفاده از الگوریتم‌های الگوریتم‌های تکاملی مبتنی بر جمعیت می‌باشد.

۳- تخمین بر اساس ماکزیمم کردن تابع چگالی احتمال

پسین

از دیدگاه ماکزیمم کردن تابع احتمال پسین مسئله تخمین به صورت زیر است [۱۴ و ۱]:

$$\hat{x}_t^{MAP} = \arg \max_x p(x_t | Y_t), \quad (11)$$

که با توجه به روابط (۳) داریم:

$$\hat{x}_t^{MAP} = \arg \max_x p(x_t | Y_t), \\ = \arg \max_x p(z_t | x_t, u_{t-1}, Y_{t-1}) p(x_t | x_{t-1}, u_{t-1}, Y_{t-1}), \quad (12) \\ = \arg \max_x p(z_t | x_t) p(x_t | x_{t-1}, u_{t-1}) p(x_{t-1} | Y_{t-1}).$$

با تعریف $p(x_0)$ به صورت:

$$p(x_0) = p(x_{t-1} | Y_{t-1}), \quad (13)$$

و جایگذاری (۱۳) در (۱۲) داریم:

$$\hat{x}_t^{MAP} = \arg \max_x \prod_{i=1}^t p(z_i | x_i) \prod_{i=1}^t p(x_i | x_{i-1}, u_{i-1}) p(x_0), \quad (14)$$

که این ماکزیمم‌سازی باید با توجه به مدل پروسه و اندازه‌گیری باشد. به عبارت دیگر ماکزیمم‌سازی باید به گونه‌ای باشد که در شرایط معادله اندازه‌گیری و پروسه صدق کند. بنابراین از دیدگاه MAP به مسئله تخمین می‌توان مانند یک مسئله بهینه‌سازی مقید (ماکزیمم) نگاه کرد، که قیود آن مدل سیستم و اندازه‌گیری می‌باشند. به زبان ریاضی این مسئله را می‌توان به صورت زیر بیان نمود:

$$\hat{x}_t = \arg \max_{x_t} \prod_{i=1}^t p(z_i | x_i) \prod_{i=1}^t p(x_i | x_{i-1}, u_{i-1}) p(x_0), \quad (15)$$

Subject to:

$$x_{t+1} = f(x_t, u_t) + w_t \quad t=0, 1, \dots, \quad (16)$$

$$z_t = h(x_t) + v_t \quad t=0, 1, \dots,$$

که i_1, i_2, i_3 به صورت تصادفی انتخاب می‌شوند؛ به طوری که:

$$i \neq i_1 \neq i_2 \neq i_3. \quad (25)$$

i نشان‌دهنده کروموزوم i از جمعیت $C(t)$ است. برای اینکه همگرایی سریع‌تر شود، بردار هدف به صورت زیر تشکیل می‌شود:

$$u_i(t) = x_{i_1}(t) + \beta(x_{Global}(t) - x_{i_3}(t)). \quad (26)$$

این کار سبب می‌شود که ضمن حفظ پراکندگی ذرات، ذرات به سمت بهترین ذره میل کنند.

```

j* ← U(1, n_x)
J ← J ∪ {j*}
For each j ∈ {1, ..., n_x} do
  If U(0,1) < p_r and j ≠ j* then
    J ← J ∪ {j}
end
end

```

شکل (۱): الگوریتم تشکیل بردار J .

۵- ایجاد فرزند $x'_i(t)$ با اعمال اپراتور پرش

ژن‌های فرزند $x'_i(t)$ به صورت زیر تعیین می‌شوند:

$$x'_{ij}(t) = \begin{cases} u_{ij}(t) & j \in J, \\ x_{ij}(t) & j \notin J, \end{cases} \quad (27)$$

که J نشان‌دهنده ژن‌های کروموزوم i ام است که تحت اپراتور پرش قرار می‌گیرند. روش‌های مختلفی برای ایجاد J وجود دارد که در این مقاله از الگوریتم داده شده در شکل ۱ استفاده شده است و

۶- در صورتی که تابع معیار فرزند $x'_i(t)$ از $x_i(t)$ بهتر باشد $x'_i(t)$ به نسل بعد منتقل خواهد شد و در غیر این صورت خود $x_i(t)$ به نسل بعد منتقل خواهد شد.

۵- الگوریتم بهینه‌سازی اجتماع ذرات

الگوریتم اجتماع ذرات برای نخستین بار توسط کندی^۳ و ابرهات^۴ معرفی گردیده است [۲۱-۲۲]. همانند سایر الگوریتم‌های جمعیتی، الگوریتم اجتماع ذرات از مجموعه‌ای

$$\hat{x}_t = \underset{x_t}{\text{Max}}(\log p_v(z_t | x_t) + \log p_w(x_t | x_{t-1}, u_{t-1})). \quad (22)$$

Subject to:

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (23)$$

$$z_t = h(x_t) + v_t.$$

همان‌طور که از این معادلات مشخص است، بهینه‌سازی این مسئله با استفاده از روش‌های کلاسیک مشکل است. در این مقاله برای حل این مسئله از الگوریتم‌های تکاملی تفاضلی و اجتماع ذرات استفاده شده است.

۴- الگوریتم تکامل تفاضلی

تکامل تفاضلی یک الگوریتم تکاملی است که به‌وسیله Storm و Price در سال ۱۹۹۵ پیشنهاد شده است [۲۰]. الگوریتم تفاضل تکاملی شبیه به سایر الگوریتم‌های تکاملی به ویژه الگوریتم ژنتیک^۱ (GA) از اپراتورهای تکاملی استفاده می‌کند. مهم‌ترین تمایز الگوریتم تفاضل تکاملی با سایر روش‌های تکاملی در این است که این الگوریتم از اطلاعات فاصله و جهت جمعیت فعلی برای هدایت پروسه جستجو استفاده می‌کند. اپراتور برش^۲ مهم‌ترین اپراتور در الگوریتم تکامل تفاضلی است. این اپراتور در ابتدا برای ایجاد بردار هدف و سپس برای تولید فرزند استفاده می‌شود. به طور خلاصه این الگوریتم دارای مراحل زیر است [۲۰]:

۱- مقداردهی اولیه پارامترهای کنترل β و p_r ,

که این پارامترها نیز می‌توانند به صورت تطبیقی تنظیم شوند و پارامتر β در بازه $[0, \infty)$ انتخاب می‌شود،

۲- ایجاد جمعیت اولیه $C(0)$

جمعیت اولیه به صورت تصادفی انتخاب می‌شود و تعداد ژن‌ها در هر کروموزوم بستگی به تعداد متغیرهای تابع معیار بستگی دارد،

۳- محاسبه تابع معیار

برای هر کروموزوم $(x_i(t))$ از جمعیت $(C(t))$ تابع معیار حساب می‌شود،

۴- تشکیل بردار هدف $u_i(t)$

بردار هدف برای هر کروموزوم با اعمال اپراتور پرش به صورت زیر تشکیل می‌شود:

$$u_i(t) = x_{i_1}(t) + \beta(x_{i_2}(t) - x_{i_3}(t)), \quad (24)$$

3- Kennedy

4- Eberhart

1- Genetic Algorithm

2- Mutation Operator

اولین جمله این معادله $p_v(z_t | x_t)$ با توجه به اندازه‌گیری $z_{t,i}$ و اندازه‌گیری تخمین زده شده $\hat{z}_{t,i}$ برای هر اندازه‌گیری به صورت زیر محاسبه می‌شود:

$$p_v(z_{t,i} | x_t) = \frac{1}{(2\pi\sigma_v^2)^{1/2}} e^{-\frac{1}{2} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{\sigma_v^2}}, \quad (31)$$

که در اینجا، σ_v^2 کواریانس نویز اندازه‌گیری و به صورت زیر است:

$$E[v_k v_k^T] = R = \sigma_v^2. \quad (32)$$

از این رو، اگر تعداد N_s اندازه‌گیری وجود داشته باشد، خواهیم داشت:

$$p_v(z_t | x_t) = \prod_{i=0}^{N_s} p(z_{t,i} | x_t) = \prod_{i=0}^{N_s} \frac{1}{(2\pi\sigma_v^2)^{1/2}} e^{-\frac{1}{2} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{\sigma_v^2}}. \quad (33)$$

برای محاسبه دومین جمله $p_w(x_i | x_{i-1}, u_{i-1})$ با توجه به مدل پروسه مطابق معادله (۱) داریم:

$$p_w(x_i | x_{i-1}, u_{i-1}) = \frac{1}{\sqrt{|P|(2\pi)^n}} e^{-\frac{1}{2}(x_i - \hat{x}_i)Q^{-1}(x_i - \hat{x}_i)^T}, \quad (34)$$

که در این معادله، \hat{x}_i به صورت زیر است:

$$\hat{x}_i = f(\hat{x}_{i-1}, u_{i-1}), \quad (35)$$

و کواریانس نویز پروسه است:

$$E[w_k w_k^T] = Q. \quad (36)$$

با توجه به اینکه مقادیر ثابت در کمینه‌سازی تأثیری ندارند. تابع هدف به صورت زیر کاهش پیدا می‌کند:

$$\tilde{f}_0(x_t) = \sum_{i=0}^{N_s} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{2\sigma_v^2} + \frac{1}{2}(x_t - \hat{x}_t)Q^{-1}(x_t - \hat{x}_t)^T. \quad (37)$$

بنابراین، به طور خلاصه مسئله تخمین به مسئله بهینه‌سازی مقید زیر تبدیل می‌شود:

$$\text{Min}_{x_t} \tilde{f}_0(x_t), \quad (38)$$

Subject to

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (39)$$

$$z_t = h(x_t) + v_t. \quad (40)$$

برای بهینه‌سازی مسئله فوق با استفاده از الگوریتم تکامل تفاضلی (DE) ابتدا جمعیت اولیه‌ای متشکل از کروموزوم‌ها

از پاسخ‌های ممکن استفاده می‌کند که این پاسخ‌ها تا زمانی که پاسخ‌های بهینه یافت شود و یا شرایط پایان یافتن الگوریتم محقق گردد، به حرکت خود در فضای جستجو ادامه می‌دهند.

در الگوریتم اجتماع ذرات دو عامل اساسی بر روی کارایی الگوریتم نتیجه مستقیم دارد، الگوریتم و اندازه جمعیت. ضرایب الگوریتم مجموعه‌ای از اعداد حقیقی هستند که در معادله الگوریتم ظاهر شده و مسیر حرکت ذرات را در فضای جستجو معین می‌نمایند. در روش‌های متداول این ضرایب عموماً به صورت ثابت و یا متغیر می‌باشند. در الگوریتم PSO ذرات در فضای جستجو حرکت کرده و هر ذره بهترین تجربه شخصی خود را اعم از بردار موقعیت یا سرعت حفظ می‌نماید و کل گروه ذرات نیز بهترین تجربه جمعی را حفظ می‌کنند. بردار سرعت در هر لحظه بر اساس سه عامل سرعت در گام قبل، بهترین تجربه شخصی و بهترین تجربه جمعی به روز شده و پس از آن موقعیت هر ذره بر اساس این سرعت جدید به روز می‌گردد. به طور کلی اگر $\vec{x}_i(t)$ نشان دهنده موقعیت ذره P_i در فضای جستجو در لحظه t باشد، موقعیت P_i با افزودن سرعت $\vec{v}_i(t)$ به موقعیت فعلی به صورت زیر تغییر می‌کند [۲۱]:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t), \quad (28)$$

$$\vec{v}_i(t) = \vec{v}_i(t-1) + c_1 r_1 (\vec{P}_{bi} - \vec{x}_i(t-1)) + c_2 r_2 (\vec{P}_g - \vec{x}_i(t-1)), \quad (29)$$

که در آن، $\vec{v}_i(t)$ بردار سرعت در گام t ، c_1 و c_2 مقادیر مثبت و r_1 و r_2 اعدادی تصادفی که به صورت نرمال در بازه $[0,1]$ تولید می‌شوند. پارامترهای \vec{P}_{bi} و \vec{P}_g به ترتیب نشان‌دهنده موقعیت بهترین تجربه شخصی و بهترین تجربه جمعی می‌باشند.

۶- الگوریتم تخمین مبتنی بر بهینه‌سازی تکامل تفاضلی

همان‌طور که در بخش‌های قبل گفته شد، هدف این است که بر اساس مدل اندازه‌گیری و مدل پروسه، تابع هدف زیر بهینه شود:

$$\tilde{f}_0(x_t) = \log p_v(z_t | x_t) + \log p_w(x_t | x_{t-1}, u_{t-1}). \quad (30)$$

آنگاه کلیه کروموزوم‌های نسل فعلی با توجه به معادلات پروسه سیستم انتقال یافته و نسل جدید به وجود می‌آید، که این همان عمل پیش‌بینی در فیلتر بیز می‌باشد. با این تفاوت که در روش بیز تنها یک جواب موجود بود ولی در اینجا مجموعه‌ای از جواب‌ها وجود دارند. شکل ۲ فلوجارت کل الگوریتم را نشان می‌دهد.

۷- الگوریتم تخمین مبتنی بر اجتماع ذرات

همان‌طور که در قسمت قبل دیده شد مسئله تخمین به صورت مسئله بهینه‌سازی مقید زیر است:

$$\text{Min}_x \left(\sum_{i=0}^{N_s} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{2\sigma_v^2} + \frac{1}{2} (x_t - \hat{x}_t) Q^{-1} (x_t - \hat{x}_t)^T \right), \quad (44)$$

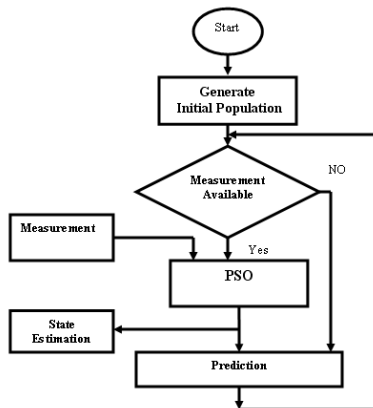
Subject to

$$x_{t+1} = f(x_t, u_t) + w_t, \quad (45)$$

$$z_t = h(x_t) + v_t. \quad (46)$$

در این قسمت، برای بهینه‌سازی مسئله فوق از الگوریتم اجتماع ذرات استفاده شده است. برای این منظور، ابتدا جمعیت اولیه‌ای متشکل از کروموزوم‌ها برای موقعیت‌ها و سرعت‌ها تشکیل می‌شود که تعداد ژن‌های برابر تعداد حالت‌هایی است که باید تخمین زده شوند. تابع معیار برای ارزیابی کروموزوم‌ها همان تابع هدف به کار برده شده در الگوریتم تفاضلی به صورت زیر است. شکل ۳ فلوجارت این الگوریتم را نشان می‌دهد.

$$\tilde{f}_{PSO}(x_t) = \sum_{i=0}^{N_s} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{2\sigma_v^2} + \frac{1}{2} (x_t - \hat{x}_t) Q^{-1} (x_t - \hat{x}_t)^T. \quad (47)$$



شکل (۳): الگوریتم فیلتر مبتنی اجتماع ذرات.

تشکل می‌شود، که تعداد ژن‌های آن بستگی به تعداد حالت‌هایی دارد که باید تخمین زده شوند. در دومین گام از الگوریتم پیشنهادی اپراتورهای پرش DE بر روی جمعیت عمل کرده و برای هر یک از کروموزوم‌ها بردار هدف زیر تشکیل می‌شود:

$$u_i(t) = x_{i_1}(t) + \beta(x_{i_2}(t) - x_{i_3}(t)). \quad (41)$$

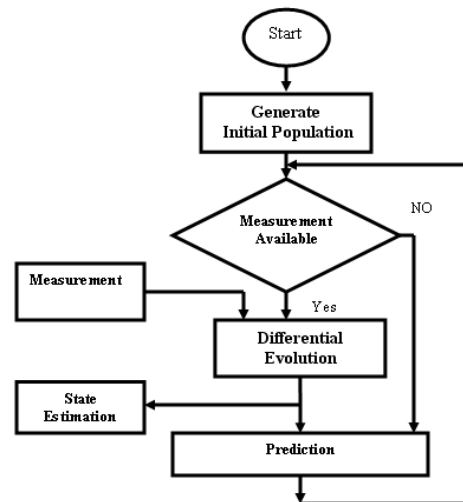
سپس، ژن‌های فرزند $x'_i(t)$ تحت اپراتور پرش به صورت زیر تعیین می‌شوند:

$$x'_{ij}(t) = \begin{cases} u_{ij}(t) & j \in J, \\ x_{ij}(t) & j \notin J, \end{cases} \quad (42)$$

که J نشان‌دهنده ژن‌های کروموزوم i ام است که تحت اپراتور پرش قرار می‌گیرند و بدین ترتیب فرزندان $x'(t)$ تولید می‌شوند. در سومین مرحله مقادیر کروموزوم‌ها ارزیابی می‌شود. در صورتی که تابع معیار فرزند $x'_i(t)$ از $x_i(t)$ بهتر باشد $x'_i(t)$ به نسل بعد منتقل خواهد شد و در غیر این صورت خود $x_i(t)$ به نسل بعد منتقل خواهد شد. تابع معیار برای ارزیابی کروموزوم‌ها همان تابع هدف بوده و به صورت زیر است:

$$\tilde{f}_{DE}(x_t) = \sum_{i=0}^{N_s} \frac{(z_{t,i} - \hat{z}_{t,i})^2}{2\sigma_v^2} + \frac{1}{2} (x_t - \hat{x}_t) Q^{-1} (x_t - \hat{x}_t)^T. \quad (43)$$

وقتی که تابع معیار در نسل فعلی از حدی کوچک‌تر شد در این صورت بهترین کروموزوم از نسل فعلی به عنوان بهترین تخمین در زمان فعلی می‌باشد.



شکل (۲): الگوریتم فیلتر مبتنی تکامل تفاضلی.

$$Cov(e(k)) = E(e(k) - \bar{e})^2, \quad \bar{e} = E(e(k)). \quad (53)$$

اگر به جای عملگر E از $\frac{1}{k} \sum_{i=1}^k$ استفاده شود در این صورت کوواریانس با استفاده از داده‌های تا لحظه k ، به صورت زیر تخمین زده می‌شود:

$$Cov(e_k) = \frac{1}{k} \sum_{i=1}^k (e(k) - \bar{e}(k))^2, \quad (54)$$

$$\bar{e}(k) = \frac{1}{k} \sum_{i=1}^k e(k).$$

برای ارزیابی عملکرد فیلتر پیشنهادی ابتدا حالتی در نظر می‌گیریم که ماتریس‌های کواریانس نویز پروسه و اندازه‌گیری به صورت زیر باشد:

$$R_k = 1^2 \quad Q_k = 5^2. \quad (55)$$

همچنین، تعداد ذرات در فیلتر ذره‌ای، تکامل تفاضلی و اجتماع ذرات بزرگ ($N=100$) انتخاب شده است. شکل‌های ۴-۷ نتایج به دست آمده از فیلتر پیشنهادی، فیلتر کالمن توسعه یافته و فیلتر ذره‌ای را نشان می‌دهد. در هر شکل حالت واقعی، حالت تخمین زده شده، خطای تخمین و انحراف استاندارد (2σ) رسم شده است. انحراف استاندارد خطا از روش مونت کارلو با ۵۰ بار اجرا به دست آمده است. همان‌طور که از شکل‌های ۴-۷ مشاهده می‌شود عملکرد فیلترهای مبتنی بر تکامل تفاضلی و اجتماع ذرات به مراتب بهتر از عملکرد فیلتر ذره‌ای و فیلتر کالمن توسعه یافته است. نکته قابل توجه دیگر در این شکل‌ها این است که فیلتر کالمن توسعه یافته ضعیف‌تر از بقیه عمل کرده است، که به دلیل غیر خطی بودن بالای سیستم است. غیر خطی بودن بالای سیستم سبب می‌شود که تقریب خطی سیستم غیرخطی بد شود و در نتیجه تابع توزیع احتمال پسین $p(x_t | Y_t)$ از حالت گوسی بودن بسیار منحرف و سبب بد شدن تخمین‌ها می‌شود. به دلیل اینکه در فیلتر پیشنهادی از الگوریتم‌هایی تکاملی که مبتنی بر تکرار در فضای جستجو هستند استفاده شده است. از این رو در ابتدا به نظر می‌رسد که هزینه زمانی افزایش خواهد یافت. اما در کاربرد ما (مسئله تخمین) الگوریتم تکاملی نیاز به تعداد تکرار زیادی ندارد. این بدان دلیل است که در مسئله تخمین، کروموزوم‌ها یا ذرات جمعیت اولیه حول مقدار واقعی (با توجه به کواریانس اولیه) مقدار دهی می‌شوند و از طرف دیگر فضای جستجو در هر

مطابق این فلوجارت در صورتی که اندازه‌گیری در دسترس باشد، تابع معیار برای هر یک از کروموزوم‌ها حساب می‌شود. وقتی که تابع معیار در نسل فعلی از حدی کوچک‌تر شد در این صورت بهترین کروموزوم از نسل فعلی به عنوان بهترین تخمین در زمان فعلی می‌باشد. سپس کلیه کروموزوم‌های نسل فعلی با توجه به معادلات پروسه سیستم انتقال می‌یابند.

۸- شبیه‌سازی و نتایج

به منظور ارزیابی فیلتر پیشنهادی، عملکرد آن با فیلتر کالمن توسعه یافته و فیلتر ذره‌ای برای یک سیستم غیرخطی مینا^۱ مقایسه شده است. میزان غیرخطی بودن سیستم زیاد می‌باشد و به همین جهت در مقالات زیادی به عنوان معیار تست جهت ارزیابی عملکرد فیلترهای پیشنهادی استفاده شده است [۵۲]. معادلات سیستم غیرخطی مینا عبارت است از:

$$x_k = f_k(x_{k-1}, k) + v_k, \quad (48)$$

$$y_k = \frac{x_k^2}{20} + w_k, \quad (49)$$

که در آن، تابع $f_k(x_{k-1}, k)$ به صورت زیر است:

$$f_k(x_{k-1}, k) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k). \quad (50)$$

همچنین، v_k و w_k به ترتیب نویزهای اندازه‌گیری و پروسه‌اند که به صورت نویز سفید با میانگین صفر و کواریانس‌های زیر می‌باشند:

$$E[w_k w_j^T] = \begin{cases} Q & k=j \\ 0 & k \neq j \end{cases}, \quad (51)$$

$$E[v_k v_j^T] = \begin{cases} R & k=j \\ 0 & k \neq j \end{cases},$$

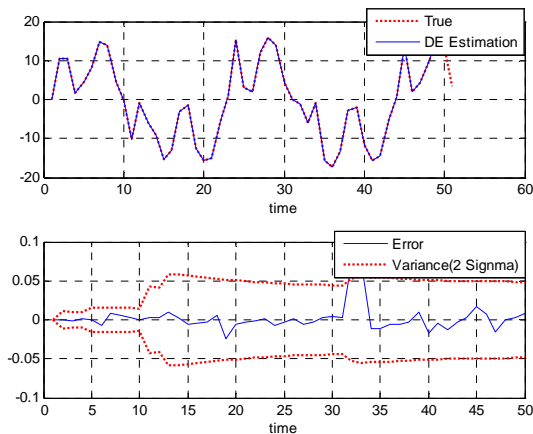
که در این صورت، روابط زیر صادق است:

$$p(x_k | x_{k-1}) = N(x_k; f_k(x_{k-1}, k), Q), \quad (52)$$

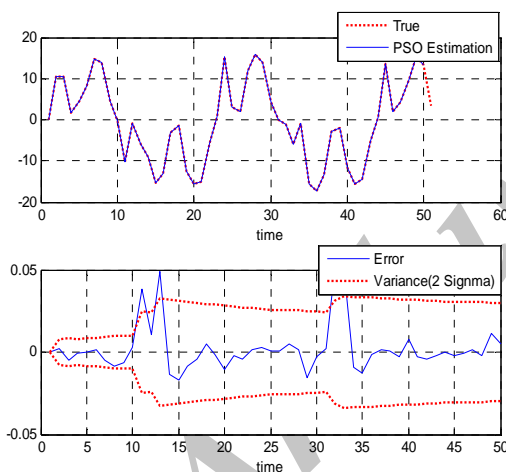
$$p(y_k | x_k) = N(y_k; \frac{x_k^2}{20}, R).$$

برای ارزیابی انحراف معیار خطای تخمین‌ها (2σ تخمین) کواریانس خطا به صورت عددی محاسبه شده است. برای این منظور کواریانس خطا $e(k)$ به صورت زیر تعریف می‌شود:

حالت عملکرد فیلتر ذره‌ای به مراتب بدتر از عملکرد فیلتر مبتنی بر تکامل تفاضلی و اجتماع ذرات نسبت به حالت قبل (۱۰۰ ذره) شده است.



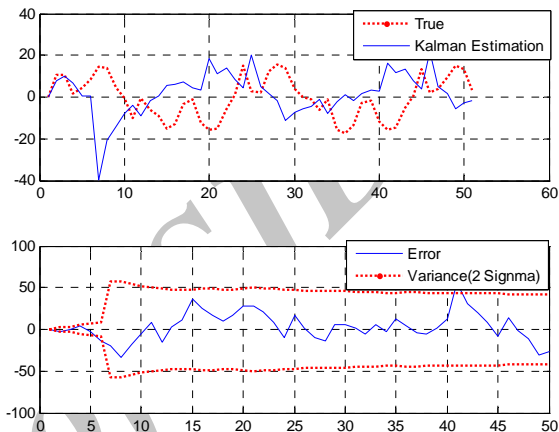
شکل (۶): فیلتر مبتنی بر تکامل تفاضلی (N=100).



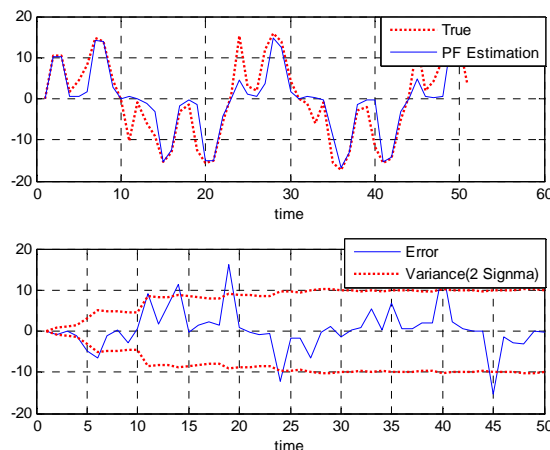
شکل (۷): فیلتر مبتنی بر اجتماع ذرات (N=100).

همچنین، عملکرد فیلتر پیشنهادی با ۱۰ ذره در این حالت بهتر از عملکرد فیلتر ذره‌ای با صد ذره است. این موضوع نشان می‌دهد که حساسیت فیلتر ذره‌ای به تعداد ذرات نسبت به روش پیشنهادی زیادتر است. در صورتی که تعداد ذرات از حدی کوچک‌تر شود، فیلتر ذره‌ای ناپایدار می‌شود. علت این موضوع این است که در این حالت ذرات فیلتر ذره‌ای نمی‌توانند به درستی تابع چگالی احتمال پسین را تخمین بزنند و در نتیجه مسئله فقر نمونه و تباهی‌گی رخ

لحظه یک ناحیه کوچک حول حالت در زمان $t-1$ (x_{t-1}) است. بنابراین یکی از مهم‌ترین خصوصیات فیلتر پیشنهادی این است که در تعداد Epoch کم به خوبی عمل می‌کند. در اینجا تعداد Epoch پنج در نظر گرفته شده است (چون تنها یک حالت داریم حتی در تعداد Epoch کمتر از ۵ هم به خوبی و با همین دقت جواب می‌دهد).

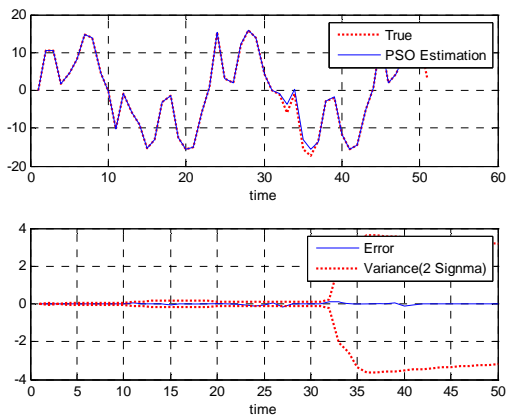


شکل (۴): فیلتر کالمن توسعه یافته (EKF).



شکل (۵): فیلتر ذره‌ای (N=100).

حال حالتی بررسی می‌شود که تعداد ذرات در فیلتر ذره‌ای، تکامل تفاضلی و اجتماع ذرات کوچک انتخاب (N=10) شود. در اینجا ماتریس‌های کواریانس نویز پروسه و اندازه‌گیری مشابه حالت قبل در نظر گرفته شده است. همان‌طور که از شکل‌های ۱۰-۸ مشاهده می‌شود در این



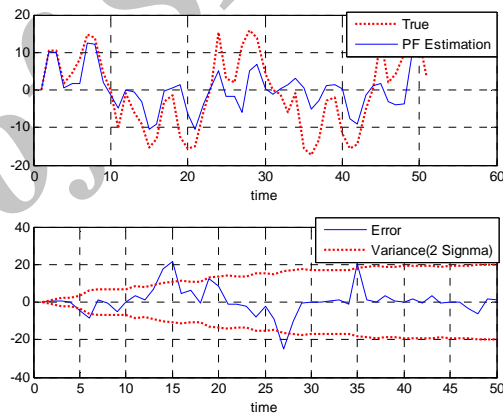
شکل (۱۰): فیلتر مبتنی بر اجتماع ذرات ($N=10$).

یکی دیگر از مزایای روش پیشنهادی نسبت به فیلتر ذره‌ای و فیلتر کالمن توسعه‌یافته، برتری عمل کرد آن در حالتی است که قدرت نویز پروسه خیلی بیشتر از اندازه‌گیری بالا باشد. شکل‌های ۱۱-۱۴ برای حالتی است که ماتریس‌های کواریانس نویز پروسه و اندازه‌گیری به صورت زیر در نظر گرفته شده باشد:

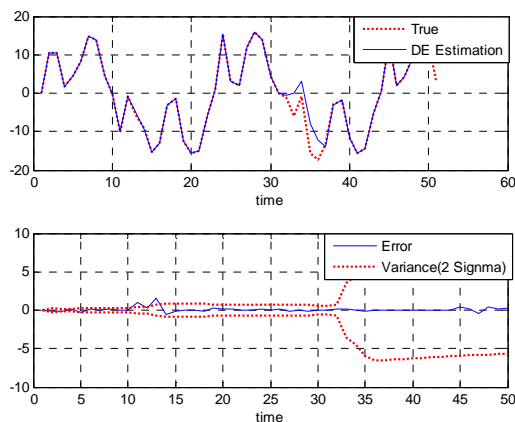
$$R_k = 2^2 \quad Q_k = 10^2. \quad (56)$$

نتایج نشان می‌دهد که عمل کرد روش پیشنهادی در این حالت مشابه عمل کرد آن در حالت ($R_k = 1^2$, $Q_k = 5^2$) است. با این حال عمل کرد فیلتر ذره‌ای نسبت به حالت ($R_k = 1^2$, $Q_k = 5^2$) خراب‌تر شده است. با توجه به اینکه در این حالت تابع پیشین پهن‌تر از تابع درست‌نمای است بنابراین به تعداد کمتری از ذرات وزن بالایی اختصاص می‌یابد و بقیه ذرات وزن ناچیزی دارند. این مسئله سبب می‌شود که ذرات تباه‌شده شوند. در صورت رخ دادن این پدیده تلاش زیادی برای به‌روزرسانی ذراتی می‌شود که سهمشان در تقریب تابع چگالی احتمال پسین تقریباً ناچیز است. از طرفی پدیده تباه‌شدگی سبب خواهد شد که بعد از مرحله نمونه-برداری مجدد^۱ تنوع میان ذرات از بین برود و سبب پدیده‌های بنام فقر نمونه شود. در مجموع این مسائل سبب خواهد شد که دقت تخمین‌های فیلتر ذره‌ای کاهش یابد.

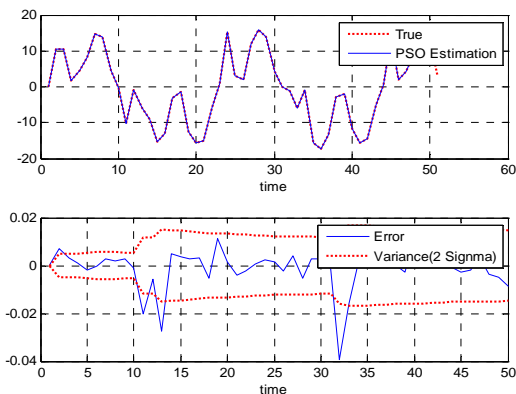
خواهد داد. با این حال، روش پیشنهادی در این حالت پایدار است و تخمین‌های آن از دقت خوبی برخوردار هستند. این به دلیل این است که در روش پیشنهادی ذرات طوری حرکت می‌کنند که در مکان‌هایی که احتمال تابع چگالی احتمال پسین بالا است قرار گیرند. در نتیجه به دلیل قرار گرفتن ذرات در مکان‌های مناسب توان تابع چگالی احتمال پسین را به خوبی تخمین زد. بنابراین، در مجموع روش پیشنهادی نیاز به تعداد ذرات کم دارد و این یکی دیگر از دلایلی است که حجم محاسبات فیلتر پیشنهادی کم است. نکته قابل توجه دیگر در این شکل‌ها این است که دقت الگوریتم مبتنی بر اجتماع ذرات بهتر از سایر روش‌ها است. و این به دلیل این است که PSO سریع‌تر ذرات را به سمت نقطه بهینه سوق می‌دهد.



شکل (۸): فیلتر ذره‌ای ($N=10$).



شکل (۹): فیلتر مبتنی بر تکامل تفاضلی ($N=10$).



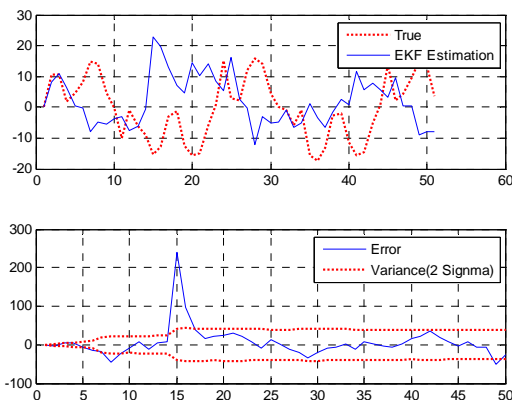
شکل (۱۴): فیلتر مبتنی بر اجتماع ذرات (N=100).

برای بررسی پایداری روش پیشنهادی نسبت به سایر فیلترها، به بررسی سازگاری^۱ آن می‌پردازیم. مسئله سازگاری یک از مسائل مهم در طراحی فیلترها می‌باشد. متغیر تصادفی x_k را در نظر بگیرید در صورتی که فیلتر میانگین \hat{x}_k و کورایانس \hat{P}_k را بدست آورد در این صورت فیلتر سازگار است در صورتی که شرایط زیر صادق باشد:

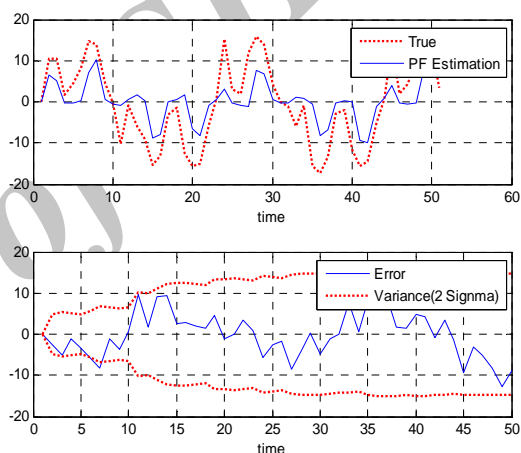
$$E(x_k - \hat{x}_k) = 0, \tag{۵۷}$$

$$E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] = \hat{P}_k.$$

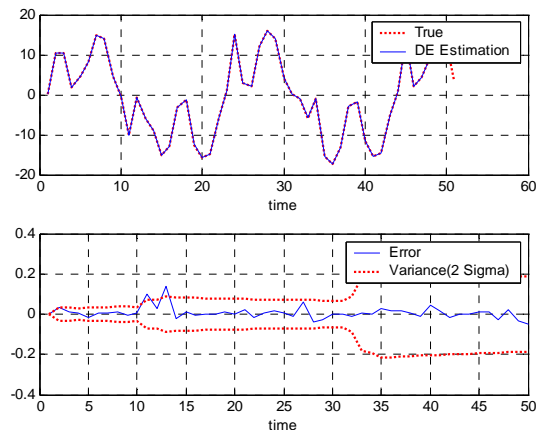
شرط اول، دلالت بر بدون بایاس بودن فیلتر دارد. شرط دوم بیان می‌دارد که برای اینکه فیلتر سازگار باشد باید کورایانس خطای محاسبه شده توسط فیلتر با کورایانس واقعی خطا برابر باشد. به عبارت دیگر فیلتر باید خاصیت تطبیق کورایانس را برآورده کند. برای سازگاری فیلتر لازم است این دو شرط همزمان با هم برقرار شوند. بررسی کاملی از انواع روش‌های تست سازگاری یک فیلتر در مرجع [۱] موجود است که برای مطالعه بیشتر می‌توان به آن‌ها مراجعه نمود. به طور ایده‌آل، برای اندازه‌گیری اینکه آیا یک فیلتر سازگار است، می‌توان تخمین‌هایش را با تابع چگالی احتمال بدست آمده از فیلتر بیزین ایده‌آل مقایسه کرد. وقتی تابع چگالی احتمال^۲ (PDF)



شکل (۱۱): فیلتر کالمن توسعه یافته (EKF).



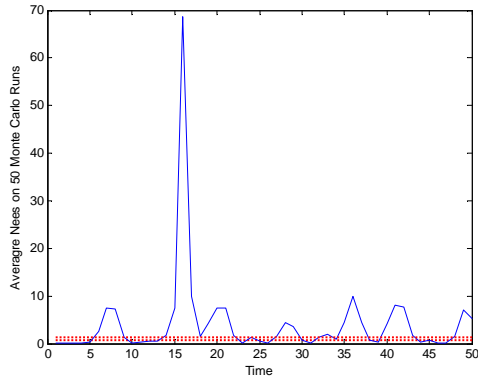
شکل (۱۲): فیلتر ذره‌ای (N=100).



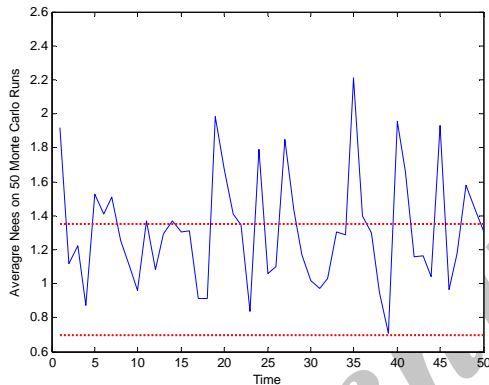
شکل (۱۳): فیلتر مبتنی بر تکامل تفاضلی (N=100).

1- Consistency
2- Probability Density Function

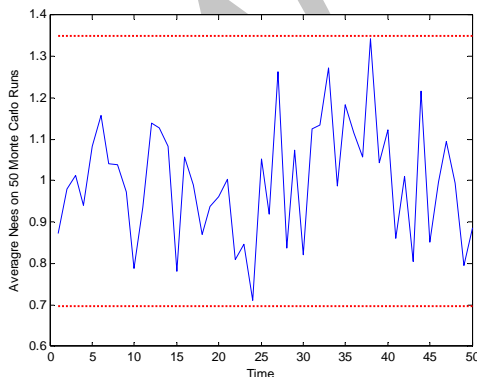
پهن تر از تابع درستنمای می باشد بنابراین به تعداد کمتری از ذرات وزن بالایی اختصاص می یابد و بقیه ذرات وزن ناچیزی دارند. این مسئله سبب کم شدن تنوع ذرات و در نتیجه کاهش سازگاری فیلتر می شود.



شکل (۱۵): سازگاری فیلتر کالمن توسعه یافته.



شکل (۱۶): سازگاری فیلتر ذره ای (N=100).



شکل (۱۷): سازگاری فیلتر مبتنی بر تفاضل تکاملی (N=100).

واقعی در دسترس نباشد، اما حالت واقعی x_k معلوم باشد در این صورت از تخمین نرمالیزه شده مربع خطا^۱ (NEES) برای مشخص کردن عملکرد سازگاری فیلتر استفاده می شود:

$$\varepsilon_k = (x_k - \hat{x}_k)^T \hat{P}_k^{-1} (x_k - \hat{x}_k). \quad (58)$$

یک اندازه از سازگاری فیلتر با تخمین متوسط NEES از روی M بار اجرای مونت کارلوی بدست می آید. تحت این فرض که فیلتر سازگار است و گوسین خطی تقریبی ε_k دارای توزیع χ^2 با درجه آزادی $\dim(x_k)$ است آنگاه مقدار متوسط ε_k وقتی M به سمت بی نهایت میل کند به سمت ابعاد حالت به صورت زیر میل خواهد می کند:

$$E(\varepsilon_k) = \dim(x_k). \quad (59)$$

سازگاری فیلتر با اجرای چندین اجرای مونت کارلو و محاسبه متوسط NEES ارزیابی می شود. برای M بار اجرای مونت کارلو متوسط NEES به صورت زیر محاسبه می شود:

$$\bar{\varepsilon}_k = \frac{1}{M} \sum_{i=1}^M \varepsilon_{i_k}. \quad (60)$$

در اینجا بایستی سازگاری فیلتر پیشنهادی، فیلتر کالمن و فیلتر ذره ای در حالتی که ماتریس های کواریانس نویز پروسه و اندازه گیری به صورت زیر باشد بررسی شود:

$$R_k = 2^2, \quad Q_k = 10^2, \quad (61)$$

به دلیل اینکه تعداد حالت های سیستم خطی مورد نظر (بعد حالت) یک است بنابراین برای $M = 50$ ، ناحیه احتمال ۹۵٪ برای $\bar{\varepsilon}_k$ محدود به بازه $[0.69 \ 1.35]$ است. شکل های ۱۸-

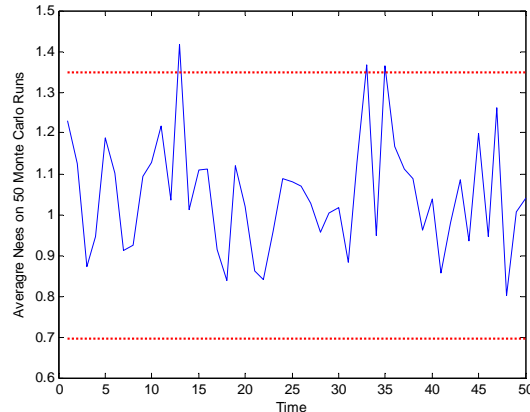
۱۵ سازگاری فیلتر کالمن توسعه یافته، فیلتر ذره ای، فیلتر مبتنی بر تکامل تفاضلی و فیلتر مبتنی بر اجتماع ذرات را نشان می دهند. نتایج نشان که در مقایسه با فیلتر کالمن توسعه یافته و فیلتر ذره ای روش پیشنهادی از سازگاری بهتری برخوردار است. همان طور که ملاحظه می شود فیلتر کالمن توسعه یافته، سازگاری ضعیف تری نسبت به سایر روش ها دارد که به دلیل غیرخطی بودن بالای سیستم است. این موضوع سبب می شود که تابع توزیع احتمال پسین از حالت گوسی بودن منحرف و سبب بد شدن تخمین ها و در نتیجه سازگاری ضعیف تر شود. در فیلتر ذره ای به دلیل اینکه تابع پیشین

1- Normalized Estimation Error Squared (NEES)

ذره‌ای واگرا خواهد شد. حال آنکه فیلتر پیشنهادی در این حالت به خوبی عمل می‌کند. یکی دیگر از امتیازات فیلتر پیشنهادی این است که در مقایسه با فیلتر کالمن توسعه‌یافته، فیلتر ذره‌ای از سازگاری بهتری برخوردار است. بنابراین در مجموع، روش پیشنهادی بر فیلتر کالمن توسعه‌یافته و فیلتر ذره‌ای برتری دارد.

۱۰- مراجع

1. Kalman, R., "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME J. Basic Eng., Series. D, No. 82, pp. 34-45, 1960.
2. Fox, D., "KLD-Sampling: Adaptive Particle Filters", Int. J. Robotics Research, Vol. 22, No. 12, pp. 985-1003, 2003.
3. Boers, Y., "On the Number of Samples to be Drawn in Particle Filtering", IEEE Colloquium on Target Tracking: Algorithms and Applications Ref. No. 1999/090, 1999/215, pp. 5/1- 5/6, 1999.
4. Ristic, B., Arulampalam, S., and Gordon, N., "In Beyond Kalman Filter: Particle Filters Tracking Application", Artech House Publishers, Boston, Ch. 1-3, pp. 3-65, 2004.
5. Simon, D., "Optimal State Estimation Kalman, H_∞ and Non-linear Approaches", John Wiley and Sons, New Jersey, 2006.
6. Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Non-linear/Non-gaussian Bayesian Tracking", IEEE Trans. Signal Process., Vol. 50, No. 2, pp. 174-188, 2002.
7. Kramer, S.C. and W.sorenson, H., "Recursive Bayesian Estimation, Using Piece-Wise Constant Approximations", Automatica, Vol. 24, No. 6, pp. 789-801, 1988.
8. Doucet, A., Gordon, N., and V. Krishnamurthy, "Particle Filters for State Estimation of Jump Markov Linear Systems", IEEE Trans. Signal Processing, Vol. 49, No. 3, pp. 613-624, 2001.
9. Zhou, J. and Pei, F., "Non-linear State Estimation, Using Adaptive Particle Filter", World Cong. on Intelligent Control and Automation, 2008.
10. Doucet, A., Freitas, N., and Gordon, N., "Sequential Monte Carlo Methods in Practice", Springer-Verlag, New York, 2001.
11. Gordon, N., Salmond, D., and Smith, A.F.M., "Novel Approach to Non-linear and Non-Gaussian Bayesian State Estimation", Radar and Signal Processing, IEE Proceedings F, Vol. 140, No. 2, pp. 107-113, 1993.
12. Park, S., Hwang, J., Kim, E., and Kang, H., "A New Evolutionary Particle Filter for the Prevention



شکل (۱۸): سازگاری فیلتر مبتنی بر اجتماع ذرات (N=100)

۹- نتیجه‌گیری

در این تحقیق، مسئله تخمین سیستم‌های غیرخطی به‌ویژه سیستم‌های با میزان غیرخطی بالا مورد مطالعه قرار گرفت. به دلیل اینکه در این سیستم‌ها تابع توزیع احتمال پسین از حالت گوسی بودن بسیار منحرف می‌شود. بنابراین، تخمین زده شده توسط فیلتر کالمن توسعه‌یافته خراب می‌شود. از این روی برای تخمین این سیستم‌ها از فیلترهای ذره‌ای باید استفاده نمود. اما مشکل فیلتر ذره‌ای این است که دقت فیلتر به شدت وابسته به تعداد ذرات و نیز نویز پروسه و اندازه‌گیری می‌باشد. در حالتی که تعداد ذرات بزرگ و نامعینی‌های نویز پروسه و اندازه‌گیری نزدیک به هم باشد (سنسورهای اندازه‌گیری خیلی دقیق نباشند)، عملکرد فیلتر ذره‌ای خوب است و برعکس. برای حل این مشکلات در این مقاله به مسئله تخمین سیستم‌های غیر خطی به‌صورت یک مسئله بهینه‌سازی مقید نگریسته شده و در ادامه یک روش جدید برای تخمین حالت‌های سیستم بر اساس روش‌های تکامل تفاضلی و اجتماع ذرات ارائه شده است. یکی از مهم‌ترین خصوصیات روش پیشنهادی این است که وابستگی آن به تعداد ذرات کم است. همچنین، دقت حالت‌های تخمین‌های زده شده توسط فیلتر پیشنهادی در حالتی که سنسورهای اندازه‌گیری دقیق باشد (نامعینی‌های نویز پروسه بیشتر از نامعینی‌های نویز اندازه‌گیری باشد) حتی با تعداد ذرات کم بسیار خوب است حال آنکه دقت فیلتر ذره‌ای در این حالت خراب می‌شود و در صورتی که تعداد ذرات کم باشد فیلتر

- of Sample Impoverishment,” IEEE Trans. on Evolutionary Computation, Vol. 13, No. 4, 2009.
13. Higuchi, T., “Monte Carlo Filter Using the Genetic Algorithm Operators”, J. Statistics Computation and Simulation, Vol. 59, No. 1, pp.1-23, 1997.
 14. Bar-Shalom, Y., Li, X.R., and Kirubarajan, T., “Estimation with Applications to Tracking and Navigation”, John Wiley and Sons, New York, 2001.
 15. Merwe, R., Doucet, A., Freitas, N., and Wan, E., “The Unscented Particle Filter,” Technical Report CUED/F INFENG/TR 380, Cambridge University, Engineering Department, 2000.
 16. Kwok, N.M., Zhou, W., Dissanayake, G., and Fang, G., “Evolutionary Particle Filter: Resampling from the Genetic Algorithm Perspective”, IEEE 2005 Int. Conf. on Intelligent Robots and Systems, pp. 2935–2940, 2005.
 17. Park, S., Hwang, J., Rou, K., and Kim, E., “A New Particle Filter Inspired by Biological Evolution: Genetic Filter”, The World Academy of Sci. Eng. and Tech., Vol. 21, 2007.
 18. Uosaki, K., Kimura, Y., and Hatanaka, T., “Non-linear State Estimation by Evolution Strategies Based Particle Filters”, Cong. Evol. Comput., Vol. 3, pp. 884–890, 2004.
 19. Clapp, T. and Godsill, S., “Improvement Strategies for Monte Carlo Particle Filters”, In Sequential Monte Carlo Method in Practice, New York; Springer, 2001.
 20. Engelbrecht, A.P., “Computational Intelligence An Introduction”, 2th Ed., John Wiley & Sons, 2007.
 21. Clerc, M. and Kennedy, J., “The Particle Swarm Explosion, Stability, and Convergence in a Multidimensional Complex Space”, IEEE Trans. on Evolutionary Computation, Vol. 6, No. 1, 2002.
 22. Kennedy, J. and Eberhart R.C., “Particle Swarm Optimization”, IEEE Int. Conf. on Neural Networks, IV , Piscataway, NJ: IEEE Service Center, 1942-1948, 1995.

Archive