Nonlinear filtering toolbox for continuous stochastic systems with discrete measurements

Jaroslav Švácha, Miroslav Šimandl, Ondřej Straka and Miroslav Flídr

Research Centre Data, Algorithms and Decision Making &
Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia in Pilsen
Czech Republic

Outline

1. Nonlinear estimation for continuous systems with discrete measurements
2. The objective
3. Nonlinear filtering toolbox for Continuous-Discrete systems
4. Example
5. Conclusion

Problem formulation

Consider multivariate nonlinear stochastic Continuous-Discrete system

$$d\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t)dt + \boldsymbol{G}(t)d\boldsymbol{w}(t) \quad \ldots \text{Itô stochastic diff. equation(SDE)}$$
$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k, t_k) + \boldsymbol{v}_k$$

$\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ . . . non-measurable state $\boldsymbol{w}(t) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}dt)$

$\boldsymbol{z}_k \in \mathbb{R}^{n_z}$ . . . measurement $\boldsymbol{v}_k \in \mathbb{R}^{n_z}$ . . . measurement white noise

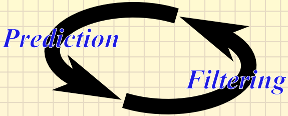
- ✓ Both noises are mutually independent and they are also independent of the known initial state $\boldsymbol{x}_0$ pdf $p(\boldsymbol{x}_0)$.
- ✓ The vector mappings $\boldsymbol{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, $\boldsymbol{h} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ and matrix $\boldsymbol{G}(t)$ are known

The aim: to the estimate the non-measurable state $x_k \triangleq \boldsymbol{x}(t_k)$

The pdf's $p(\boldsymbol{x}_k|\boldsymbol{z}^k)$ and $p(\boldsymbol{x}(t)|\boldsymbol{z}^k)$ for $t \in I_{k,k+1} \triangleq (t_k, t_{k+1}]$ are sought!

$\boldsymbol{z}^k \triangleq [\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_k]$ . . . set of measurements

General Solution



Solution of the filtering problem found using Bayesian approach

$$p(\mathbf{x}_k|\mathbf{z}^k) = \frac{p(\mathbf{x}_k|\mathbf{z}^{k-1})\, p(\mathbf{z}_k|\mathbf{x}_k)}{\int p(\mathbf{x}_k|\mathbf{z}^{k-1})\, p(\mathbf{z}_k|\mathbf{x}_k) d\mathbf{x}_k}, \qquad p(\mathbf{x}_0|\mathbf{z}^{-1}) = p(\mathbf{x}_0)$$

Solution of the prediction problem given by Fokker-Planck equation (FPE)

$$\frac{\partial p(\mathbf{x}(t)|\mathbf{z}^k)}{\partial t} = -\frac{\partial p(\mathbf{x}(t)|\mathbf{z}^k)}{\partial \mathbf{x}(t)} \boldsymbol{f}(\mathbf{x}(t),t) - p(\mathbf{x}(t)|\mathbf{z}^k)\,\mathrm{tr}\left(\frac{\partial \boldsymbol{f}(\mathbf{x}(t),t)}{\partial \mathbf{x}(t)}\right) + \frac{1}{2}\,\mathrm{tr}\left(\boldsymbol{Q}(t)\frac{\partial^2 p(\mathbf{x}(t)|\mathbf{z}^k)}{\partial \mathbf{x}^2(t)}\right)$$

with initial condition $p(\mathbf{x}_k|\mathbf{z}^k)$

Solutions of the Bayesian and Fokker-Planck equations

Exact solutions - valid only for special class of systems

- Kalman-Bucy filter - linear Gaussian system
- Daum filters - exponential family pdf's

Approximate local methods - approximation using Taylor expansion

- Extended Kalman-Bucy filter
- Iterated Kalman-Bucy filter

Approximate global methods

- analytical approach - $p(v_k)$ and $p(x_0)$ considered as Gaussian mixtures
- numerical approach - employs numerical solution of the FPE
- simulation approach - employs numerical simulation of SDE

The objective: To design toolbox facilitating easy estimator design and testing for continuous-discrete systems

What criteria should the toolbox meet?

- ✔ to be highly modular, easily extensible and user friendly
- ✔ to provide conditional prediction and filtering pdf's
- ✔ to be build in MATLAB environment

Which tasks should be provided by the toolbox?

- ✔ complete description of the continuous-discrete system
- ✔ simulation of the system
- ✔ choice and application of the suitable estimator
- ✔ easy extensibility with new estimators

Features of the Nonlinear Filtering Toolbox or Continuous-Discrete systems (NFTCD)

Advantages of presented framework

- takes advantage of Matlab **object oriented** programming features
- estimators provide conditional probability density functions
- provides means for easy control of the whole estimation process
- easy addition of new estimators
- support for various SDE solvers

Structure of NFTCD

- probability density function classes
- system classes
- estimator classes
- auxiliary classes

Classes necessary for description of the system

Probability density function classes

- all random quantities represented as objects of corresponding pdf class
- generic class defining mandatory interface of all pdf classes and making them distinguishable as pdf's within toolbox
- pdf classes provide methods such as:
  - resetting and reading of pdf parameters,
  - evaluation of pdf in arbitrary point of state space,
  - generating of random samples, . . .

Classes provided for system creation and handling

- several classes for various type of system - (Non)Linear (Non)Gaussian with (Non)Additive noises (`nlngacd`, `nlgacd`, `lngacd`, `lgacd`)
- two classes required by system class constructor
  - `sdeito` - Itô stochastic differential equation
  - `oe` - measurement equation

Estimator classes

Main task of the estimator classes

The estimator classes essentially implement algorithms necessary to obtain $p(\boldsymbol{x}_k|\boldsymbol{z}^k)$ and $p(\boldsymbol{x}(k)|\boldsymbol{z}^k)$.

Features of the general class `estimatorcd`

- its virtual methods sets the interface of actual estimator classes
- provides methods `estimate` and `extestimate` that controls the whole estimation process $\Rightarrow$ the designer of the estimator doesn't need to care
- `estimatorcd` stores the data of multistep operations in dynamical list
- the lists can hold arbitrary content, however, they are primarily used to store conditional pdf's
- implements commonly used methods (e.g. Ricatti equation) $\Rightarrow$ decreases redundancy and makes possible easy future improvements

Estimator classes

Estimators currently implemented in NFTCD

Method	NFTCD class
Kalman-Bucy filter	`kalmancd`
Extended Kalman-Bucy filter	`extkalmancd`
Iterating Kalman-Bucy filter	`itekalmancd`
Second order Kalman-Bucy filter	`seckalmancd`
Gaussian sums filter	`gsmcd`
Particle filter	`pfcd`
Unscented Kalman filter	`nfcd`

How to implement new estimator?

- firstly choose the type of conditional pdf
- create three necessary method of `estimatorcd` child class (i.e. the class constructor, the `filtering` and `prediction` methods)
- `filtering` and `prediction` methods hand over their results

Example of NFTCD usage

Consider the following continuous stochastic process $x(t)$ observed at discrete time instants t_k ($t_0 = 0s$, $t_1 = 0.1s$, $t_2 = 0.2s, \ldots$)

$$dx(t) = (x(t) - 0.4x^2(t))dt + dw(t)$$
$$z_k = x_k^2 + v_k$$

The description of stochastic quantities

$$p(x_0|z^{-1}) = p(x_0) = 0.5\mathcal{N}(x_0 : -2, 1) + 0.5\mathcal{N}(x_0 : 1, 2)$$
$$p(v_k) = \mathcal{N}(v_k : 0, 1)$$

Estimation procedure using NFTCD (1/3)

- definition of the random variables

```
alpha0=[.5;.5]; x0={[-2]; [1]}; P0={[1]; [2]};
px0 = gspdf(alpha0,x0,P0);
pw = gpdf(0,1); pv = gpdf(0,1)
```

Example of NFTCD usage (continuation)

Estimation procedure using NFTCD (2/3)

- creation of the model

```
sys_sde = sdeito('x-0.4*x^2','1', ...
                          'x','w',pw,px0);
sys_oe = oe('x^2','x','v',pv);
system = nlngacd(sys_sde,sys_oe);
```

- the simulation of the system

```
scheme.type = 'euler';
scheme.T = 2;
scheme.Delta = 0.001;
time = [0:0.1:scheme.T];
[system,x,xDelta,z,winc,v] = ...
         simulate(system, scheme, time);
```

- choise and creation of the estimators

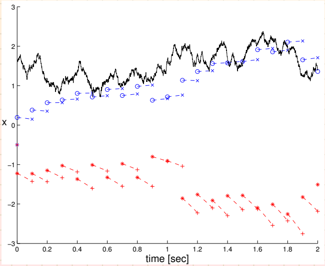
```
f1 = gsmcd(system,time,0);
f2 = extkalmancd(system,time,0);
```

Example of NFTCD usage (continuation)

Estimation procedure using NFTCD (3/3)

- estimation process itself

```
scheme.type = 'euler'; scheme.N = 10;
pred_num = 100;
[est1,predall1,pred1] = ...
          extestimate(f1,z,pred_num,scheme);
[est2,predall2,pred2] = ...
          extestimate(f2,z,pred_num,scheme);
```



Concluding remarks

Current contribution of the NFTCD

- provides all necessary tools for estimator design, testing and employment
- the toolbox is easily extensible thanks to object oriented approach
- includes set of basic estimators

Future directions

- implementation of additional estimators
- merging with Nonlinear Filtering Toolbox designed only for discrete time systems