Autonomous vehicles (IMARO)
Statistical Signal Processing and Estimation Theory (SIP, EPICO)
Lab work

1 Random variables

Please see the appendix “Matlab, Octave” of the book.

1.1 A few univariate distributions

We consider a zero-mean unit-variance random variable, in the 3 cases below:

- X is normally distributed;
- X is uniformly distributed;
- X is driven by a Gaussian mixture whose both components have the same probability, the same variance; the means are $\pm m$ (necessarily, the variance of each component is $1 - m^2$); we take $m = 0.95$.

In all cases, with $N = 100$ and $N = 4000$:

a) Generate N realizations.[1]

b) Plot the normalized histogram of these N realizationsand plot the probability density function on the same figure.

c) Estimate the mean and the standard deviation (`mean, std`).

d) Plot the N realizations as an independent random signal.

1.2 Joint distribution

We consider a zero-mean bivariate normal random variable $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$, where X_1 and X_2 have variances σ_1^2, σ_2^2, and correlation coefficient ρ. Use $\sigma_1 = 2$, $\sigma_2 = 5$, $\rho = 0,9$.

- Generate $N = 200$ realizations of this random variable.
- Estimate the mean and the variance (`mean, cov`).
- plot these realizations together with the 91 % confidence ellipses.

2 Kalman filter

A DC motor is driven by the input voltage $u(t)$. The angular position of the rotor $\theta(t)$ is measured with an incremental encoder (precision $L = 512$ angles per lap) which provides the measure $y(t)$ of $\theta(t)$. $\Omega(t)$ is the angular velocity ($\Omega(t) = \dot{\theta}(t)$). To perform a velocity control, we have to estimate online $\theta(t)$ and $\Omega(t)$, from $u(t)$ and $y(t)$.

2.1 Input voltage

The input voltage $u(t)$ is a zero-mean square wave with period $\Delta = 100$ ms, and peak-to-peak amplitude $A = 0{,}1$ V. This signal is sampled with sample time $T_s = 1$ ms. Create a `MatLab` function which provides this sampled input for a duration D: `u = inputvoltage(D,A,Delta,Ts)`, where `u` is a column vector which contains the sampled input (`square`).

---

[1] To generate N realizations of a zero-mean unit-variance random variable:

Gauss	: `x = randn(N,1);`
Uniform	: `x = 2*sqrt(3)*(rand(N,1)-0.5);`
Symmetric Gaussian mixture	: `m=0.95; x=randn(N,1)*sqrt(1-m*m)+m; k=find(rand(N,1)>0.5); x(k) = x(k)-2*m;`

2.2 System modeling and simulation

With the state vector $x(t) = \begin{bmatrix} \theta(t) \\ \Omega(t) \end{bmatrix}$, we get the state-space representation:

$$\begin{cases} \dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{T} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{G}{T} \end{bmatrix} u \\ \theta = \begin{bmatrix} 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \end{bmatrix} u \end{cases}$$

The input-output signals are sampled with the sample time T_s, The input $u(t)$ is constant between 2 sampling times. Thus, the continuous-time above can be sampled without any approximation using the step invariance method (or zero-order-hold method, "zoh") [2]. For all integer n, and all function f, we note $f_n = f(n\,T_s)$.
The measure y_n provided by the incremental encoder is a quantization of the actual angular position θ_n.[3]

a) Write a function `[y,x] = simulate(u,G,T,Ts,L,x1)`, where `y` is a column matrix (same size as `u`) which contains the evolution of the output y_n, `x` is a 2-columns matrix which contains the evolution of the state vector; `x1` is the initial state vector).

b) Test the simulator with $G = 50$ rad.s^{-1}.V^{-1} et $T = 20$ ms.

2.3 Kalman filter

w_n is the quantization noise of the incremental encoder, r is its variance ($y_n = \theta_n + w_n$). To take into account modeling errors, we assumes that the actual input is $u_n + v_n$, where v_n is a white noise with variance q, independent of w_n. The goal is to estimate X_n. The motor is initially stopped, but there is no information on the initial angular position.

a) Write the equations of the Markov model of the system with input u_n and output y_n.

b) Propose a value of r (it is usual to model a quantization error as a uniform random variable).

c) Propose an initialization of the prior information $\hat{X}_{1/0}$ and propose a value for its variance $P_{1/0}$.

d) Write the Kalman filter[4] and the stationary Kalman filter[5], that is two functions `xe = kal(y, u, G, T, Ts, L, x1_0, P1_0, q)`, where `xe` is a 2-columns matrix which contains the evolution of the state vector estimation.

2.4 Simulations

Compare the estimation of the position and the velocity given by both filters for different values of q (use the initialization $\hat{\theta}_{1/0} = \theta_1 \pm 0.05$) in the following cases:

- the model of the system is perfect: $\begin{cases} G_{\text{actual}} = G_{\text{filter}} = 50 \text{ rad.s}^{-1}.\text{V}^{-1} \\ T_{\text{actual}} = T_{\text{filter}} = 20 \text{ ms} \end{cases}$
- the model of the system is rough: $\begin{cases} G_{\text{actual}} = G_{\text{filter}} = 50 \text{ rad.s}^{-1}.\text{V}^{-1} \\ T_{\text{actual}} = 20 \text{ ms} \\ T_{\text{filter}} = 25 \text{ ms} \end{cases}$

"actual" index if for the value used in the simulation, "filter" index is for the value used in the Kalman filter.

---

[2] For a continuous-time system with the state space representation:

$$\begin{cases} \dot{x}(t) = A\,x(t) + B\,u(t) \\ y(t) = C\,x(t) + D\,u(t) \end{cases}$$

the step invariance sampling with sample time T_s writes (see `MatLab` function `c2dm`):

$$\begin{cases} x_{n+1} = \tilde{A}\,x_n + \tilde{B}\,u_n \\ y_n = C\,x_n + D\,u_n \end{cases} \text{ with } \begin{cases} \tilde{A} = e^{A\,T_s} \\ \tilde{B} = \int_0^{T_s} e^{A\,\tau}\,B\,d\tau \end{cases}$$

[3] With L angles per lap, the precision is $2\,\pi/L$ rad. With `MatLab`: `y = round(teta*L/2/pi)*2*pi/L`

[4] With time-varying Kalman gain, page 47 of the book.

[5] With constant gain obtained with function `dlqe`, page 48 of the book.