

Estimação e Identificação - Trabalho sobre o Filtro de Kalman

Paulo Lopes dos Santos

Nov. 2023

1 - Escrever e testar, em Matlab, uma função que simule o sistema

$$\begin{aligned}x(t+1) &= Ax(t) + Bu(t) + q(t) \\ y(t) &= Cx(t) + Du(t) + r(t)\end{aligned}$$

em que

$$\begin{aligned}u(t) &\in \mathbb{R}^m \\ y(t), r(t) &\in \mathbb{R}^\ell \\ x(t), q(t) &\in \mathbb{R}^n \\ A &\in \mathbb{R}^{n \times n} \\ B &\in \mathbb{R}^{n \times m} \\ C &\in \mathbb{R}^{\ell \times n} \\ D &\in \mathbb{R}^{\ell \times m}\end{aligned}$$

As entradas desta função são

- Parâmetros do sistema
 - *sys* - Objeto idss com os parâmetros do sistema
 - x_0 - vector coluna ou linha de dimensão n (estado inicial do sistema)
- Sinais
 - u - Matriz de dimensão $N \times m$ ou $m \times N$ em que cada coluna (ou linha) armazena um canal de entrada e cada linha (ou coluna) um instante de tempo (são simulados N instantes de tempo)
 - q - Matriz de dimensão $N \times n$ ou $n \times N$.
 - r - matriz de dimensão $N \times \ell$ ou $\ell \times N$.

As saídas:

- y - Matriz $N \times \ell$ ou $\ell \times N$ (cada coluna um canal de saída e cada linha um instante de tempo ou cada linha é um canal de saída e cada coluna um instante de tempo).
- x - Matriz $N \times n$ ou $n \times N$ (cada coluna uma variável de estado e cada linha um instante de tempo ou cada linha uma variável de estado e cada coluna um instante de tempo).

Esta função deve chamar-se *simSS* e deve ser guardada no ficheiro "SimSS.m".
Nota: *idss* é uma classeda toolbox de Identificação de Sistemas. Os objetos desta classe podem criados pelo comando:

```
>> sys=idss(A,B,C,D)
```

2 - Simule o sistema com parâmetros guardados no ficheiro "T1-Dados.mat" com entrada $u(t)$ ruído branco binário com valor médio nulo e variância unitária, $q(t)$ e $r(t)$ realizações independentes de ruído branco gaussianos com média nulas e matrizes de covariância $Q = \mathbf{E}\{q(t)q^T(t)\}$ e $R = \mathbf{E}\{r^2(t)\}$, respectivamente (os valores de Q e R encontram-se no ficheiro *T1-Dados.mat*) e com $x(0) = \begin{bmatrix} -500 \\ 100 \end{bmatrix}$

Sugestão: Utilize o comando do Matlab *idinput* para gerar $u(t)$ (para ver como funciona este comando digite *help idinput* no Matlab)

3- Escreva e teste, em MATLAB, uma função que implemente o previsor e filtro de Kalman As variáveis entradas desta função devem ser:

- sinais- objeto iddata com os sinais de entrada e de saída do sistema
- sys - Objeto idss com os parâmetros do modelo verdadeiro
- xp0 - Estimativa do estado inicial.
- P0 - Covariância do erro da estimativa inicial do estado inicial.
- ruido - Estrutura com os as matrizes de covariância do ruído:
 - ruido.Q= $\mathbf{E}\{q(t)q^T(t)\}$, matriz covariância do ruído de estado.
 - ruido.S= $\mathbf{E}\{q(t)r^T(t)\}$, covariância entre os ruídos de estado e de medição.
 - R= $\mathbf{E}\{r(t)r^T(t)\}$, covariância do ruído de medição.

As saídas:

- xp - matriz das previsões do estado calculadas pelo previsor de Kalman.
- xf - matriz das estimativas de estado do filtro de Kalman.
- yp - Previsões da saída.
- yf - Saída estimada pelo filtro de Kalman.
- K - Ganhos do previsor de Kalman.
- Kf - Ganhos do filtro de Kalman.
- P - Matrizes de covariância dos erros da previsão dos estados.
- Pf - Matrizes de covariância dos erro do filtro de Kalman.

Nota:

A função deve ter o nome de **NonStatKalman** e deve estar no ficheiro **Non-StatKalman.m**

3 - Utilize o previsor de Kalman para estimar o estado do sistema simulado na tarefa 2. Compare as estimativas de estado do previsor e do filtro com as de um observador de Luenberger com valores próprios $\Lambda_1 = \Lambda_2 = 0.8$.