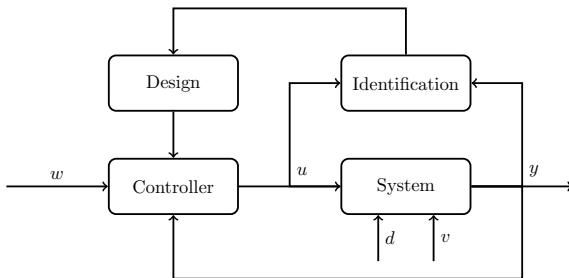


- ① ARX models
- ② ARX estimation
- ③ ARMAX estimation
- ④ ARMAX identification
- ⑤ ARMAX control
- ⑥ ARMAX adaptive control
- ⑦ Systems and control theory
- ⑧ Stochastic systems + SS estimation
- ⑨ SS estimation + control
- ⑩ SS control
- ⑪ SS parameter estimation
- ⑫ SS nonlinear estimation
- ⑬ SS nonlinear control



Today's Agenda



- Least-squares estimation
- Maximum likelihood estimation
- Cramér-Rao lower bound

Least-squares estimation

System

$$x_{t+1} = A(\theta)x_t + B(\theta)u_t + G(\theta)v_t, \quad (1)$$

$$y_t = C(\theta)x_t + D(\theta)u_t + F(\theta)e_t, \quad (2)$$

Stochastic vectors

$$x_0 \sim N(m_0(\theta), P_0(\theta)), \quad v_t \sim N(0, I), \quad e_t \sim N(0, I) \quad (3)$$

Least-squares parameter estimation problem

State prediction

$$\hat{x}_{t+1} = A\hat{x}_t + Bu_t, \quad \hat{x}_0 = m_0 \quad (4)$$

Output prediction

$$\hat{y}_t = C\hat{x}_t + Du_t \quad (5)$$

Residuals

$$\epsilon_t = y_t - \hat{y}_t \quad (6)$$

Least-squares parameter estimation problem

$$\hat{\theta} = \arg \min_{\theta} J_N(\theta; Y_N), \quad J_N(\theta; Y_N) = \frac{1}{2} \sum_{t=0}^N \epsilon_t^T \epsilon_t \quad (7)$$

This problem is nonlinear in the parameters, θ , and the solution must be approximated numerically, e.g., using Matlab's `fmincon`.

Matlab's fmincon

Syntax

```

x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(____)
[x,fval,exitflag,output] = fmincon(____)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(____)

```

Description

Nonlinear programming solver.

Finds the minimum of a problem specified by

$$\min_x f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases} \quad (8)$$

b and beq are vectors, A and Aeq are matrices, $c(x)$ and $ceq(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $ceq(x)$ can be nonlinear functions. x , lb , and ub can be passed as vectors or matrices; see Matrix Arguments.

Link: <https://de.mathworks.com/help/optim/ug/fmincon.html>

Matlab's fmincon: Least-squares objective function

```
1  function JN = least_squares_objective_function(theta, Y, U, m0, p)
2      % Create system matrices (provided by the user)
3      [A, B, C, D] = p.create_system_matrices(theta, p);
4
5      % Initial state
6      xhatt = m0;
7
8      % Objective function
9      JN = 0;
10
11     for t = 1:N+1 % (the real t is actually 0:N)
12         % Manipulated input and output
13         ut = U(:, t);
14         yt = Y(:, t);
15
16         % Predict output and residual
17         yhatt = C*xhatt + D*ut;
18         epsilont = yt - yhatt;
19
20         % Predict state
21         xhattp1 = A*xhatt + B*ut;
22
23         % Add to objective function
24         JN = JN + 0.5*(epsilont'*epsilont);
25
26         % Update states
27         xhatt = xhattp1;
28     end
29 end
```

Maximum likelihood estimation

Likelihood

Multiplication rule

$$P(A, B) = P(A|B)P(B), \quad P(A, B|C) = P(A|B, C)P(B|C) \quad (9)$$

Probability density function

$$p(y_N, y_{N-1}, \dots, y_0 | \theta) = p(y_N | y_{N-1}, \dots, y_0, \theta) p(y_{N-1}, \dots, y_0 | \theta) \quad (10)$$

Likelihood

$$\mathcal{L}(\theta) = p(y_N, y_{N-1}, \dots, y_0 | \theta) = p(y_0 | \theta) \prod_{t=1}^N p(y_t | y_{t-1}, \dots, y_0, \theta) \quad (11)$$

Log-likelihood

$$\ln \mathcal{L}(\theta) = \ln p(y_0 | \theta) + \sum_{t=1}^N \ln p(y_t | y_{t-1}, \dots, y_0, \theta) \quad (12)$$

Likelihood for normally distributed variables

Probability density of normal distribution

$$p(y_t|y_{t-1}, \dots, y_0, \theta) = \frac{1}{\sqrt{(2\pi)^{n_y} \det P_{y,t|t-1}}} \exp\left(-\frac{1}{2}\epsilon_t^T P_{y,t|t-1}^{-1} \epsilon_t\right) \quad (13)$$

Residuals

$$\epsilon_t = y_t - \hat{y}_{t|t-1} \quad (14)$$

Logarithm of probability density of normal distribution

$$\ln p(y_t|y_{t-1}, \dots, y_0, \theta) = -\frac{n_y}{2} \ln 2\pi - \frac{1}{2} \ln \det P_{y,t|t-1} - \frac{1}{2} \epsilon_t^T P_{y,t|t-1}^{-1} \epsilon_t \quad (15)$$

Log-likelihood

$$\mathcal{L}(\theta) = -\frac{(N+1)n_y}{2} \ln 2\pi - \frac{1}{2} \sum_{t=0}^N \left(\ln \det P_{y,t|t-1} + \epsilon_t^T P_{y,t|t-1}^{-1} \epsilon_t \right) \quad (16)$$

Maximum likelihood estimation

Maximum likelihood estimation

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta) \quad (17)$$

Equivalent formulation

$$\hat{\theta} = \arg \min_{\theta} J_N(\theta), \quad J_N(\theta) = -\mathcal{L}(\theta) \quad (18)$$

Negative log-likelihood function

$$J_N(\theta) = \frac{(N+1)n_y}{2} \ln 2\pi + \frac{1}{2} \sum_{t=0}^N \left(\ln \det P_{y,t|t-1} + \epsilon_t^T P_{y,t|t-1}^{-1} \epsilon_t \right) \quad (19)$$

Key differences to least-squares objective function

- 1 Determinant of **covariance** is penalized
- 2 Residuals are weighted by the inverse of the **covariance**

Kalman filter equations

Measurement update (vectors)

$$\hat{y}_{t|t-1} = C\hat{x}_{t|t-1} + Du_t, \quad (20)$$

$$\epsilon_t = y_t - \hat{y}_{t|t-1}, \quad (21)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + \kappa_t \epsilon_t \quad (22)$$

Measurement update (matrices)

$$P_{y,t|t-1} = CP_{t|t-1}C^T + R_2, \quad (23)$$

$$P_{xy,t|t-1} = P_{t|t-1}C^T, \quad (24)$$

$$\kappa_t = P_{xy,t|t-1}P_{y,t|t-1}^{-1}, \quad (25)$$

$$P_{t|t} = P_{t|t-1} - \kappa_t P_{xy,t|t-1}^T \quad (26)$$

Time update

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t} + Bu_t, \quad (27)$$

$$P_{t+1|t} = AP_{t|t}A^T + R_1 \quad (28)$$

Matlab's fmincon: Negative log-likelihood objective function



```

1  function JN = maximum_likelihood_objective_function(theta, Y, U, m0, P0, p)
2      % Create system matrices (provided by the user)
3      [A, B, G, C, D, F] = p.create_system_matrices(theta, p);
4
5      % Initial state and covariance
6      xhattm1 = m0;
7      Pttm1    = P0;
8
9      % Objective function
10     JN = 0.5*numel(Ybar)*log(2*pi);
11
12     for t = 1:N+1 % (the real t is actually 0:N)
13         % Manipulated input and output
14         ut = U(:, t);
15         yt = Y(:, t);
16
17         % Predicted output and covariance, residual, and Kalman gain
18         yhattm1 = C*xhattm1 + D*ut;
19         Pyttm1  = C*Pttm1*C' + R2;
20         epsilon = yt - yhattm1;
21         kappat  = Pttm1*C'/Pyttm1;
22
23         % Measurement update
24         xhattt = xhattm1 + kappat*epsilon;
25         Ptt    = Pttm1 - kappat*C*Pttm1;
26
27         % Time update
28         xhattp1 = A*xhattt + B*ut;
29         Pttp1   = A*Ptt*A' + R1;
30
31         % Add to objective function
32         JN = JN + 0.5*(log(det(Pyttm1)) + epsilon'(Pyttm1\epsilon));
33
34         % Update states
35         xhattm1 = xhattp1; Pttm1 = Pttp1;
36     end
37 end

```

Cramér-Rao lower bound

Cramér-Rao lower bound

Lower bound on individual parameter variances¹

$$\text{Cov}(\hat{\theta}) \succeq F^{-1} \quad (29)$$

where $A \succeq B$ means that $A - B$ is positive semidefinite.

Fisher's information matrix

$$F_{ij} = \mathbb{E} \left[\frac{\partial J_N}{\partial \theta_i}(\theta; Y_N) \frac{\partial J_N}{\partial \theta_j}(\theta; Y_N) \right] \quad (30)$$

Typically, maximum likelihood estimators are *efficient*, which means that equality holds in the bound

$$F_{ij} = \left(\frac{\partial m_Y}{\partial \theta_i} \right)^T P_Y^{-1} \frac{\partial m_Y}{\partial \theta_j} + \frac{1}{2} \text{Tr} \left(P_Y^{-1} \frac{\partial P_Y}{\partial \theta_i} P_Y^{-1} \frac{\partial P_Y}{\partial \theta_j} \right) \quad (31)$$

¹Theorem 4.4 in the book by A. van den Bos, 2007. Parameter estimation for scientists and engineers. Wiley.

The prediction problem: Compact notation (recap)

Compact notation

$$X_N = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix}, \quad Y_N = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad U_N = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}, \quad V_N = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_N \end{bmatrix}, \quad E_N = \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_N \end{bmatrix}, \quad (32)$$

$$\Phi_{xx} = \begin{bmatrix} I \\ A \\ \vdots \\ A^N \end{bmatrix}, \quad \Gamma_{xu} = \begin{bmatrix} 0 & & \\ B & 0 & \\ \vdots & \ddots & \ddots \\ A^{N-1}B & \dots & B & 0 \end{bmatrix}, \quad \Gamma_{xv} = \begin{bmatrix} 0 & & \\ G & 0 & \\ \vdots & \ddots & \ddots \\ A^{N-1}G & \dots & G & 0 \end{bmatrix}, \quad (33)$$

$$\Phi_{yx} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^N \end{bmatrix}, \quad \Gamma_{yu} = \begin{bmatrix} D & & \\ CB & D & \\ \vdots & \ddots & \ddots \\ CA^{N-1}B & \dots & CB & D \end{bmatrix}, \quad \Gamma_{yv} = \begin{bmatrix} 0 & & \\ CG & 0 & \\ \vdots & \ddots & \ddots \\ CA^{N-1}G & \dots & CG & 0 \end{bmatrix}, \quad (34)$$

$$\Gamma_{ye} = \begin{bmatrix} F & & \\ & F & \\ & & \ddots \\ & & & F \end{bmatrix}, \quad R_V = I, \quad (35)$$

$$X_N = \Phi_{xx}x_0 + \Gamma_{xu}U_N + \Gamma_{xv}V_N, \quad V_N \sim N(0, R_V), \quad (36)$$

$$Y_N = \Phi_{yx}x_0 + \Gamma_{yu}U_N + \Gamma_{yv}V_N + \Gamma_{ye}E_N, \quad E_N \sim N(0, R_E) \quad (37)$$

*The superscript N on the matrices and the dependencies on the parameters, θ , have been omitted for brevity of notation.

Distribution of prediction

Distribution of output prediction

$$Y_N \sim N(m_Y, P_Y) \quad (38)$$

Mean

$$m_Y = \mathbb{E}[Y_N] = \Phi_{yx}m_0 + \Gamma_{yu}U_N \quad (39)$$

Covariance

$$P_Y = \text{Cov}(Y_N) = \Phi_{yx}P_0\Phi_{yx}^T + \Gamma_{yv}R_V\Gamma_{yv}^T + \Gamma_{ye}R_E\Gamma_{ye}^T \quad (40)$$

Negative log-likelihood function (multiplication rule not used)

$$J_N(\theta) = -\ln \mathcal{L}(\theta; Y_N) = \frac{(N+1)n_y}{2} \ln(2\pi) + \frac{1}{2} \ln \det P_Y \quad (41)$$

$$+ \frac{1}{2}(Y_N - m_Y)^T P_Y^{-1}(Y_N - m_Y) \quad (42)$$

Problem: $P_Y \backslash \epsilon$ might give imprecise results (and a warning)

Use an LDL factorization ($P_Y = LDL^T$)

$$LZ = \epsilon, \quad Z = DL^T X, \quad (43)$$

$$DY = Z, \quad Y = L^T X, \quad (44)$$

$$L^T X = Y \quad (45)$$

Questions?