Stochastic Adaptive Control (02421)

Lecture 5

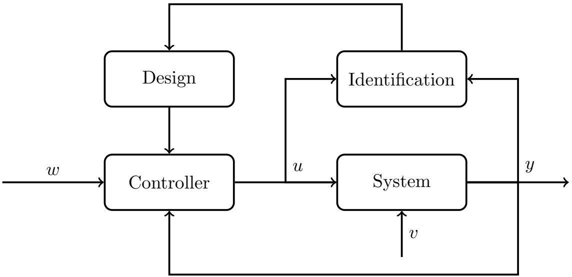
Tobias K. S. Ritschel

Assistant Professor

Section for Dynamical Systems, DTU Compute

Lecture Plan

1. ARX models
2. ARX prediction + control
3. ARX estimation
4. ARX model validation + adaptive control
5. ARMAX control
6. ARMAX estimation + adaptive control
7. Systems and control theory
8. Stochastic systems + Kalman filtering
9. SS estimation (recursive) + control
10. SS control
11. SS estimation (batch)
12. SS estimation (recursive)
13. SS nonlinear control



Today's Agenda

- MA, ARMA, and ARMAX models
- Prediction
- Generalized minimum variance (GMV) control

MA, ARMA, and ARMAX models

The Moving-Average (MA) Process

MA(n) process

$$y_t = e_t + \sum_{k=1}^{n} c_k e_{t-k}, \qquad c_0 = 1 \tag{1}$$

$\{e_t\}$ is a white-noise process (independent and Gaussian with variance σ^2)

The Moving-Average (MA) Process

Shift operator, q

$$q^{-1} y_t = y_{t-1}, \tag{2}$$

MA(n) process (compact notation)

$$y_t = C(q^{-1}) e_t, \tag{3}$$

Polynomial

$$C(q^{-1}) = 1 + \sum_{k=1}^{n} c_k q^{-k} \tag{4}$$

The Moving-Average (MA) Process

Properties of finite-order MA processes

- Always stationary
- Invertible if the zeros of C lie within the unit circle

Invertibility: The innovations can be represented as functions of past observations

The Moving-Average (MA) Process

Auto-covariance of MA(n) process

$$\gamma(k) = \begin{cases} \sigma^2\Big(c_k + c_1 c_{k+1} + \cdots + c_{n-k}c_n\Big), & |k| = 0, \dots, n, \\ 0, & |k| > n \end{cases} \tag{5}$$

Variance (constant)

$$\sigma_y^2 = \gamma(0) = \sigma^2 \left(1 + \sum_{k=1}^{n} c_k^2\right) \tag{6}$$

The ARMA Process

ARMA(m,n) process

$$y_t + \sum_{k=1}^{m} a_k y_{t-k} = e_t + \sum_{k=1}^{n} c_k e_{t-k} \tag{7}$$

$\{e_t\}$ is a white-noise process

The ARMA Process

ARMA(m,n) process (compact notation)

$$A(q^{-1})y_t = C(q^{-1})e_t \tag{8}$$

Polynomials

$$A(q^{-1}) = 1 + \sum_{k=1}^{m} a_k q^{-k}, \qquad C(q^{-1}) = 1 + \sum_{k=1}^{n} c_k q^{-k} \tag{9}$$

Transfer function: $\dfrac{C(q^{-1})}{A(q^{-1})}$

ARMA – Stationary mean

Stationary mean ($\mathbb{E}[y_t] = \bar{y}$)

$$\mathbb{E}\left[\sum_{k=0}^{m} a_k y_{t-k}\right] = \mathbb{E}\left[\sum_{k=0}^{n} c_k e_{t-k}\right], \tag{10}$$

$$\sum_{k=0}^{m} a_k \mathbb{E}[y_{t-k}] = \sum_{k=0}^{n} c_k \mathbb{E}[e_{t-k}] = 0, \tag{11}$$

$$\sum_{k=0}^{m} a_k \bar{y} = \left(\sum_{k=0}^{m} a_k\right) \bar{y} = 0 \tag{12}$$

If $A(1) = \sum_{k=0}^{m} a_k \neq 0$

$$\bar{y} = 0 \tag{13}$$

ARMA – Stationary cross-covariance

Covariance (lag 0)

$$\operatorname{Cov}\left(\sum_{k=0}^{m} a_k y_{t-k}, e_t\right) = \operatorname{Cov}\left(\sum_{k=0}^{n} c_k e_{t-k}, e_t\right) = c_0 \operatorname{Cov}(e_t, e_t) = c_0 \sigma^2 \tag{14}$$

Use that $\operatorname{Cov}(y_{t-k}, e_{t-j}) = 0$ for $k > j$

$$\operatorname{Cov}\left(\sum_{k=0}^{m} a_k y_{t-k}, e_t\right) = a_0 \operatorname{Cov}(y_t, e_t) = a_0 \gamma_{ye}(0) \tag{15}$$

Covariance (lag 0)

$$\gamma_{ye}(0) = \frac{c_0}{a_0} \sigma_\varepsilon^2 \tag{16}$$

ARMA – Stationary cross-covariance

Covariance (lag 1)

$$\mathrm{Cov}\left(\sum_{k=0}^{m} a_k y_{t-k}, e_{t-1}\right) = \mathrm{Cov}\left(\sum_{k=0}^{n} c_k e_{t-k}, e_{t-1}\right) \tag{17}$$

$$= c_1 \,\mathrm{Cov}(e_{t-1}, e_{t-1}) = c_1\sigma^2 \tag{18}$$

Use that $\mathrm{Cov}(y_{t-k}, e_{t-j}) = 0$ for $k > j$

$$\mathrm{Cov}\left(\sum_{k=0}^{m} a_k y_{t-k}, e_{t-1}\right) = a_0 \,\mathrm{Cov}(y_t, e_{t-1}) + a_1 \,\mathrm{Cov}(y_{t-1}, e_{t-1}) \tag{19}$$

$$= a_0\gamma_{ye}(1) + a_1\gamma_{ye}(0) \tag{20}$$

Covariance (lag 1)

$$\gamma_{ye}(1) = \frac{c_1}{a_0}\sigma^2 - \frac{a_1}{a_0}\gamma_{ye}(0) \tag{21}$$

Continue up to e_{t-n} to obtain $\gamma_{ye}(k)$ for $k = 0, \ldots, n$

ARMA – Stationary cross-covariance

Covariance

$$\mathrm{Cov}\left(\sum_{k=0}^{m} a_k y_{t-k}, y_{t-j}\right) = \mathrm{Cov}\left(\sum_{k=0}^{n} c_k e_{t-k}, y_{t-j}\right), \quad j = 0, \ldots, m \qquad (22)$$

Use $\gamma(-k) = \mathrm{Cov}(y_t, y_{t-k}) = \mathrm{Cov}(y_{t-k}, y_t) = \gamma(k)$ to form a linear system

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{m-1} & a_m \\ a_1 & a_0 + a_2 & a_3 & \cdots & a_m & 0 \\ a_2 & a_1 + a_3 & a_0 + a_4 & \cdots & 0 & 0 \\ a_3 & a_2 + a_4 & a_1 + a_5 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{m-1} & a_{m-2} + a_m & a_{m-3} & \cdots & a_0 & 0 \\ a_m & a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix} \begin{bmatrix} \gamma(0) \\ \gamma(1) \\ \gamma(2) \\ \gamma(3) \\ \vdots \\ \gamma(m-1) \\ \gamma(m) \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{m-1} \\ b_m \end{bmatrix} \qquad (23)$$

$$b_j = \sum_{k=j}^{n} c_k \gamma_{ye}(k-j)$$

Custom Matlab routine: trfvar.m

Advanced External Model structures

ARMAX

$$A(q^{-1})y_t = B(q^{-1})u_t + C(q^{-1})e_t \tag{24}$$

Stationary mean ($\mathbb{E}[y_t] = \bar{y}$, $u_t = \bar{u}$)

$$\mathbb{E}\left[\sum_{k=0}^{m} a_k y_{t-k}\right] = \mathbb{E}\left[\sum_{k=0}^{\ell} b_k u_{t-k} + \sum_{k=0}^{n} c_k e_{t-k}\right], \tag{25}$$

$$\sum_{k=0}^{m} a_k \mathbb{E}[y_{t-k}] = \sum_{k=0}^{\ell} b_k u_{t-k} + \sum_{k=0}^{n} c_k \mathbb{E}[e_{t-k}] = \sum_{k=0}^{n} b_k u_{t-k}, \tag{26}$$

$$\sum_{k=0}^{m} a_k \bar{y} = \left(\sum_{k=0}^{m} a_k\right)\bar{y} = \sum_{k=0}^{\ell} b_k u_{t-k} = \left(\sum_{k=0}^{\ell} b_k\right)\bar{u} \tag{27}$$

If $A(1) = \sum_{k=0}^{m} a_k \neq 0$

$$\bar{y} = \frac{B(1)}{A(1)}\bar{u} \tag{28}$$

The covariance is the same as for ARMA models

Open-loop simulation

For each time, t $(a_0 = 1)$

1. Evaluate y_t as a function of y_{t-j} for $j = 1, \ldots, n_a$, u_{t-k-j} for $j = 0, \ldots, n_b$, and e_{t-j} for $j = 0, \ldots, n_c$

$$\begin{aligned} y_t = {} & b_0 u_{t-k} + \cdots + b_{n_b} u_{t-k-n_b} + c_0 e_t + \cdots + c_{n_c} e_{t-n_c} \\ & - a_1 y_{t-1} - \cdots - a_{n_a} y_{t-n_a} \end{aligned} \tag{29}$$

Example code

```
    % Extract noise and outputs
    Yt = y(t - (1:na)); % na = numel(A) - 1;
    Ut = u(t - (0:nb)); % nb = numel(B) - 1;
    Et = e(t - (0:nc)); % nc = numel(C) - 1;

    % Compute new outputs
    y(t) = B*Ut' + C*Et' - A(2:end)*Yt';
```

Closed-loop simulation

For each time, t $(a_0 = 1)$

1. Evaluate y_t as a function of y_{t-j} for $j = 1, \ldots, n_a$, u_{t-k-j} for $j = 0, \ldots, n_b$, and e_{t-j} for $j = 0, \ldots, n_c$

$$\begin{aligned} y_t = {} & b_0 u_{t-k} + \cdots + b_{n_b} u_{t-k-n_b} + c_0 e_t + \cdots + c_{n_c} e_{t-n_c} \\ & - a_1 y_{t-1} - \cdots - a_{n_a} y_{t-n_a} \end{aligned} \tag{30}$$

2. Evaluate u_t as a function of w_{t-j} for $j = 0, \ldots, n_q$, y_{t-j} for $j = 0, \ldots, n_s$, and u_{t-j} for $j = 1, \ldots, n_r$

$$\begin{aligned} u_t = {} & \big(q_0 w_t + \cdots + \cdots q_{n_q} w_{t-n_q} - s_0 y_t - \cdots - s_{n_s} y_{t-n_s} \\ & - r_1 u_{t-1} - r_{n_r} u_{t-n_r} \big) / r_0 \end{aligned} \tag{31}$$

Closed-loop simulation

Example code

```
% Extract noise and outputs
Yt = y(t - (1:na)); % na = numel(A) - 1;
Ut = u(t - (0:nb)); % nb = numel(B) - 1;
Et = e(t - (0:nc)); % nc = numel(C) - 1;

% Compute new outputs
y(t) = B*Ut' - A(2:end)*Yt' + C*Et';

% Extract noise and outputs
Yt = y(t - (0:ns)); % ns = numel(S) - 1;
Ut = u(t - (1:nr)); % nr = numel(R) - 1;
Wt = w(t - (0:nq)); % nq = numel(Q) - 1;

% Compute new inputs
u(t) = (Q*Wt' - S*Yt' - R(2:end)*Ut')/R(1);
```

Prediction

Prediction in the ARMA Structure

ARMA process

$$A(q^{-1})y_t = C(q^{-1})e_t \tag{32}$$

Diophantine equation

$$C(q^{-1}) = A(q^{-1})G_m(q^{-1}) + q^{-m}S_m(q^{-1}) \tag{33}$$

m-step prediction based on the solution to the Diophantine equation

$$y_{t+m} = \frac{C(q^{-1})}{A(q^{-1})}e_{t+m} = G_m(q^{-1})e_{t+m} + \frac{S_m(q^{-1})}{A(q^{-1})}e_t \tag{34}$$

Prediction and error

$$\hat{y}_{t+m|t} = \frac{S_m(q^{-1})}{A(q^{-1})}e_t = \frac{S_m(q^{-1})}{A(q^{-1})}\left(\frac{A(q^{-1})}{C(q^{-1})}y_t\right) = \frac{S_m(q^{-1})}{C(q^{-1})}y_t, \tag{35}$$

$$\tilde{y}_{t+m|t} = G_m(q^{-1})e_{t+m} \tag{36}$$

$\hat{y}_t$ and $\tilde{y}_t$ are independent

This approach requires that $C(q^{-1})$ is inversely stable

Prediction in the ARMAX structure

System

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t \tag{37}$$

k is the control delay

m-step prediction

$$\hat{y}_{t+m|t} = \frac{1}{C(q^{-1})}\left(B(q^{-1})G_m(q^{-1})u_{t+m-k} + S_m(q^{-1})y_t\right), \tag{38}$$

$$\tilde{y}_{t+m|t} = G_m(q^{-1})e_{t+m} \tag{39}$$

Diophantine equation

$$C(q^{-1}) = A(q^{-1})G_m(q^{-1}) + q^{-m}S_m(q^{-1}) \tag{40}$$

The order of G_m and S_m are $m-1$ and $\max(n_a-1, n_c-m)$ and $G_m(0) = 1$

Proof of ARMAX prediction

Rewrite future output using the Diophantine equation

$$y_{t+m} = \frac{C(q^{-1})}{C(q^{-1})} y_{t+m} \tag{41}$$

$$= \frac{A(q^{-1})G_m(q^{-1}) + q^{-m}S_m(q^{-1})}{C(q^{-1})} y_{t+m} \tag{42}$$

$$= \frac{G_m(q^{-1})}{C(q^{-1})} A(q^{-1}) y_{t+m} + \frac{S_m(q^{-1})}{C(q^{-1})} y_t \tag{43}$$

Substitute system description

$$y_{t+m} = \frac{G_m(q^{-1})}{C(q^{-1})} (B(q^{-1})u_{t+m-k} + C(q^{-1})e_{t+m}) + \frac{S_m(q^{-1})}{C(q^{-1})} y_t \tag{44}$$

$$= \frac{1}{C(q^{-1})} \Big(G_m(q^{-1})B(q^{-1})u_{t+m-k} + S_m(q^{-1})y_t \Big) + G_m(q^{-1})e_{t+m} \tag{45}$$

$$= \hat{y}_{t+m|t} + \tilde{y}_{t+m|t} \tag{46}$$

Generalized minimum variance control

Generalized Minimum Variance Strategy

ARMAX model

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t \tag{47}$$

Cost function

$$J_t = \mathbb{E}\Big[\,(\tilde{y}_t - \tilde{w}_t)^2 + \rho\tilde{u}_t^2\Big] \tag{48}$$

Filtered variables

$$\tilde{y}_t = H_y(q)y_t = \frac{B_y(q^{-1})}{A_y(q^{-1})}y_t, \tag{49}$$

$$\tilde{w}_t = H_w(q)w_t = \frac{B_w(q^{-1})}{A_w(q^{-1})}w_t, \tag{50}$$

$$\tilde{u}_t = H_u(q)u_t = \frac{B_u(q^{-1})}{A_u(q^{-1})}u_t \tag{51}$$

$\rho > 0$ is a regularization parameter

$$A_y(0) = A_w(0) = A_u(0) = B_u(0) = 1 \tag{52}$$

Generalized Minimum Variance Strategy

Control law (q^{-1} is omitted for brevity)

$$\Big[A_u BG + \alpha C B_u\Big] u_t = A_u \Big[C \frac{B_w}{A_w} w_t - \frac{S}{A_y} y_t\Big], \quad \alpha = \frac{\rho}{b_0} \tag{53}$$

G and S are solutions to the Diophantine equation

$$B_y(q^{-1}) C(q^{-1}) = A_y(q^{-1}) A(q^{-1}) G(q^{-1}) + q^{-k} S(q^{-1}), \tag{54}$$

$G(0) = B_y(0)$, ord$[G] = k - 1$ and
ord$[S] = \max(n_a + n_{a_y} - 1, n_{b_y} + n_c - k)$

Note: The Diophantine equation is independent of the control filter $\frac{B_u}{A_u}$

Generalized Minimum Variance Strategy

Stationary closed-loop system

$$\Big[BA_uB_y + \alpha AB_uA_y\Big]y_t = q^{-k}\frac{B_w}{A_w}BA_uA_yw_t + RA_ye_t \tag{55}$$

$$\Big[BA_uB_y + \alpha AB_uA_y\Big]u_t = \frac{B_w}{A_w}AA_uA_yw_t - SA_ue_t \tag{56}$$

$$R = \Big[A_uBG + \alpha CB_u\Big] \tag{57}$$

We can affect the closed-loop poles using the control filter because it doesn't affect the Diophantine equation

GMV – Special cases

PZ-control

$$H_y(q) = A_m(q^{-1}), \quad H_w(q) = B_m(q^{-1}), \quad H_u(q) = 1, \quad \rho = 0 \tag{58}$$

Variant of MV_0 control

$$H_y(q) = 1, \quad H_w(q) = \frac{B_w(q^{-1})}{A_w(q^{-1})}, \quad H_u(q) = 1, \quad \rho = 0 \tag{59}$$

MV_{1a} control:

$$H_y(q) = 1, \quad H_w(q) = 1, \quad H_u(q) = 1 - q^{-1}, \quad \rho \neq 0 \tag{60}$$

MV_3 control:

$$H_y(q) = \frac{A_e(q^{-1})}{B_e(q^{-1})}, \quad H_w(q) = \frac{A_e(q^{-1})B_m(q^{-1})}{B_e(q^{-1})A_m(q^{-1})}, \quad H_u(q) = 1, \quad \rho = 0 \tag{61}$$

MV_3 control

ARMAX model

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t \tag{62}$$

Cost function

$$J_t = \mathbb{E}\left[\left(\frac{A_e(q^{-1})}{B_e(q^{-1})}y_{t+k} - \frac{A_e(q^{-1})B_m(q^{-1})}{B_e(q^{-1})A_m(q^{-1})}w_t\right)^2\right] \tag{63}$$

Control law

$$BGu_t = C\frac{A_eB_m}{B_eA_m}w_t - \frac{S}{B_e}y_t \tag{64}$$

Stationary closed-loop system

$$y_t = q^{-k}\frac{B_m}{A_m}w_t + G\frac{B_e}{A_e}e_t \tag{65}$$

$$u_t = \frac{AB_m}{BA_m}w_t - \frac{SB_e}{BA_e}e_t \tag{66}$$

Questions?