# Stochastic Adaptive Control (02421)

Lecture 1

Tobias K. S. Ritschel

Assistant Professor

Section for Dynamical Systems, DTU Compute

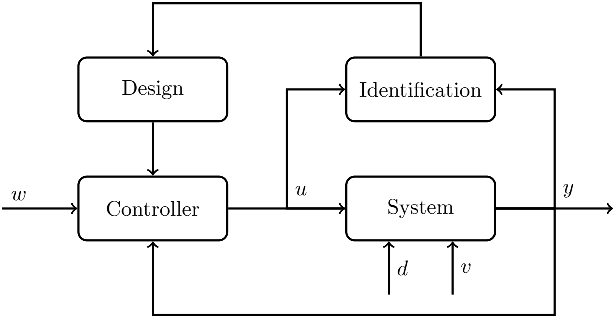
# Course Content

## Course details

- Time: Tuesday 08:15 - 12:00 (lectures and exercises)
- 5 ECTS points
- Evaluation: 3 individual reports
- Software: MATLAB (free choice)

## Course plan

- Stochastic processes and systems (ARMAX and state space models)
- Filter and control design
- System identification
- Adaptive control



# Teachers

Tobias K. S. Ritschel
Assistant Professor
Course responsible
Contact: tobk@dtu.dk
Office: 303B-052

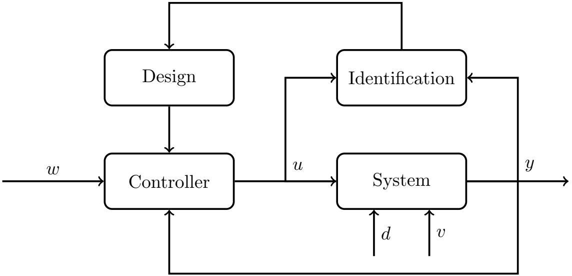
Hjalte Overgård Pind
PhD Candidate
Teaching assistant
Contact: hovpi@dtu.dk
Office: 303B-061

# You

- All of you are **MSc** students.
- A few guest students (abroad).
- Most of you are from **electrical** engineering (incl. **autonomous systems**) or **mathematical** engineering (incl. **human-oriented AI**).
- A few from **information technology**, **construction and mechanics**, and **medicine and technology**

# Lecture Plan

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. To be decided ...
12. SS nonlinear estimation
13. SS nonlinear control



# Structure of lectures

- Lectures and exercise sessions will be integrated
- Agenda + practical information
- Lecture content (1+1+1+1 model)

# Workload

Advice for this year

- If you're comfortable with the exercises, the mandatory assignments should be manageable. But consider working together with your fellow students, even though the report is individual.
- It's normal to feel stuck in this course. Therefore, ask questions!

New this year

- Harmonization of curriculum
- TCLab for testing
- More focused assignments (3 instead of 4)

# Matlab toolboxes

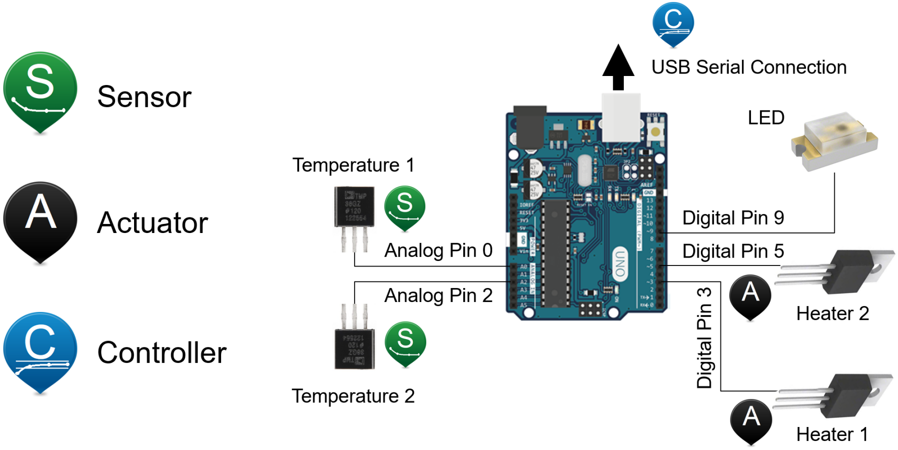
Core Matlab toolboxes

- Control toolbox
- System identification toolbox
- Optimization toolbox
- Statistics and machine learning toolbox
- Arduino Explorer App

You might need commands from these toolboxes as well

- Signal processing toolbox
- Curve fitting toolbox
- Econometrics toolbox
- Fuzzy logic toolbox

# Temperature control laboratory (TCLab)




Link: `https://apmonitor.com/pdc/index.php/Main/ArduinoTemperatureControl`

# TCLab model

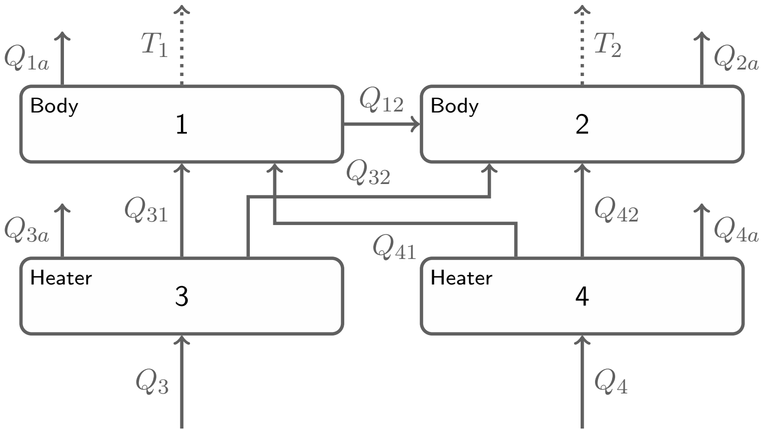



Figure: Four-compartment model of TCLab device.

# Demonstration

# Today's Agenda

Probability theory and AR(MA)X models

- Probability theory
- Auto-regressive (AR) models
- Moving average (MA) models
- Auto-regressive moving average models with exogenous inputs (ARMAX)

# Probability theory

# Stochastic scalar variables

Stochastic variable

$$X \sim \mathcal{F}(\mathbf{p}) \tag{1}$$

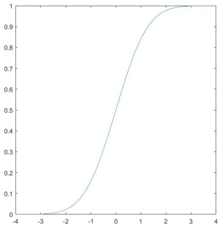
Cumulative distribution function $F_X(y)$ (cdf)

$$F_X(y) = \Pr\{X \le y\} \in [0,1], \quad \Pr\{a \le X \le b\} = F_X(b) - F_X(a) \tag{2}$$

Probability density function $f_X(z) \ge 0$ (pdf)

$$F_X(y) = \int_{-\infty}^{y} f_X(z)\,\mathrm{d}z, \tag{3}$$

$$F(-\infty) = 0, \quad F(\infty) = 1 \tag{4}$$



## Confidence Interval

$1-p$ confidence interval $CI(p)$

$$\Pr\{a \le X \le b\} = 1-p \tag{5}$$

Confidence interval based on inverse cdf

$$\Pr\{X \le a\} = p/2 \quad \text{or} \quad \Pr\{X \le b\} = 1-p/2 \tag{6}$$

$$CI(p) = [F_X^{-1}(p/2), F_X^{-1}(1-p/2)] \tag{7}$$

Use Matlab routines (or look-up tables) to compute $F_X^{-1}(p/2)$

$$X \in [m_X - \sigma_X F_X^{-1}(p/2), m_X + \sigma_X F_X^{-1}(p/2)] \tag{8}$$

Example: Let $X \sim N(10,4)$. Then, a 95% CI is

$$10 - 2\cdot 1.96 \le X \le 10 + 2\cdot 1.96 \text{ or } 6.08 \le X \le 13.92 \tag{9}$$

# Moments

For a random variable $X$

$$\textbf{Nth moment: } \mathbb{E}[X^n] = \int_\Omega x^n f(x)\,\mathrm{d}x \tag{10}$$

Moments represent certain properties of stochastic variables.

$$\textbf{Mean (1st moment): } \mathbb{E}[X] = m_x \tag{11}$$

$$\begin{aligned}\textbf{Variance (2nd central moment): } \mathrm{Var}(X) &= \mathbb{E}[(X - m_x)^2] \\ &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \sigma_x^2 \end{aligned} \tag{12}$$

# Moments - Sample Moments

Let $\{x_i\}_{i=1}^{N}$ be samples of $X$

Estimates of first and second order moments

$$\mathbb{E}[X] \approx \sum_{i=1}^{N} \frac{x_i}{N} \tag{13}$$

$$\mathrm{Var}(X) \approx \sum_{i=1}^{N} \frac{(x_i - \mathbb{E}[X])^2}{N} \tag{14}$$

Unbiased estimate of variance

$$\mathrm{Var}(X) \approx \sum_{i=1}^{N} \frac{(x_i - \mathbb{E}[X])^2}{N-1} \tag{15}$$

$$\tag{16}$$

## Probabilities: Joint probability and independence

Marginal probability that a single statement $(X \leq x)$ is true

$$\Pr\{X \leq x\} = F_X(x) \tag{17}$$

**Joint probability** that two (or more) statements are true

$$\Pr\{X \leq x, Y \leq y\} = F_{X,Y}(x,y) \tag{18}$$

Compute the **marginal distribution** from the joint distribution

$$f_X(x) = \int_{\Omega_y} f_{X,Y}(x,y)\,\mathrm{d}y \tag{19}$$

Joint distributions for **independent** variables

$$F_{X,Y}(x,y) = F_X(x)F_Y(y), \quad f_{X,Y}(x,y) = f_X(x)f_Y(y) \tag{20}$$

# Covariance

**Covariance** is a measure of how two stochastic variables varies relatively to each other

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - m_x)(Y - m_y)] \tag{21}$$

Variance is covariance between the same variable

$$\mathrm{Var}(X) = \mathrm{Cov}(X, X) \tag{22}$$

**Correlation coefficient**

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\,\mathrm{Var}(Y)}}, \quad -1 \leq \rho \leq 1 \tag{23}$$

Covariance of independent variables

$$\mathrm{Cov}(X, Y) = \rho = 0 \tag{24}$$

Note: The reverse if not true

# Conditional distribution – Bayes' theorem

Conditional probability and Bayes' theorem

$$\Pr(A|B)\Pr(B) = \Pr(A,B) = \Pr(B|A)\Pr(A), \tag{25}$$

$$\Pr\{X \le x|Y \le y\}\Pr\{Y \le y\} = \Pr\{X \le x, Y \le y\} \tag{26}$$

$$= \Pr\{Y \le y|X \le x\}\Pr\{X \le x\} \tag{27}$$

Conditional probability density function

$$f_{X|Y}(x|y)f_Y(y) = f_{X,Y}(x,y) = f_{Y|X}(y|x)f_X(x) \tag{28}$$

The same can be done for the moments if $\mathrm{Var}(X|Y) < \infty$ exists

$$\mathbb{E}[X|Y] = m_{x|y} = \int_{\Omega_x} x f_{X|Y}(x|y)\,\mathrm{d}x \tag{29}$$

$$\mathrm{Var}(X|Y) = \mathbb{E}[(X - m_{x|y})^2|Y] \tag{30}$$

# Different Distributions - Gaussian and $\chi^2$

Gaussian/normal distribution

$$X \sim N(m_x, \sigma_x^2) \tag{31}$$

$$Y = \frac{X - m_x}{\sigma_x} \sim N(0,1) \quad \text{standard Gaussian} \tag{32}$$

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{(x-m_x)^2}{2\sigma_x^2}\right) \tag{33}$$

$$F_X(x) = F_Y\left(\frac{x-m_x}{\sigma_x}\right) \tag{34}$$

$\chi^2$-distribution

$$X = \sum_{i=1}^{n} \psi_i^2 \sim \chi^2(n), \quad \psi_i \sim N(0,1), \quad \psi_i \perp \psi_j \tag{35}$$

$$f(x) = \frac{1}{\Gamma(n/2)} x^{n/2-1} \exp\left(-\frac{x}{2}\right) \tag{36}$$

$$\mathbb{E}[X] = n \quad \mathrm{Var}(X) = 2n \tag{37}$$

# Different Distributions - Gamma

Gamma distribution ($\chi^2(n) = \Gamma(n/2, 2)$)

$$X \sim \Gamma(k, \theta), \quad 0 < X < \infty \tag{38}$$

$$f_X(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} \exp\left(-\frac{x}{\theta}\right) \tag{39}$$

$$\mathbb{E}[X] = k\theta, \quad \text{Var}(X) = k\theta^2 \tag{40}$$

Gamma function

$$\Gamma(k) = \int_0^\infty t^{k-1} e^{-t} dt \tag{41}$$

$$\Gamma(k+1) = k\Gamma(k) \tag{42}$$

$$\Gamma(1) = 1, \quad \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \tag{43}$$

Erlang distribution (Gamma distribution for integer values of $k$)

$$\Gamma(k) = (k-1)!, \quad \Gamma\left(k + \frac{1}{2}\right) = \frac{(2k-1)!}{2^k}\sqrt{\pi} \tag{44}$$

# Other Related Distributions

The F-distribution

$$X = \frac{Zm}{Yn} \sim F(n, m) \tag{45}$$

$$Z \sim \chi^2(n), \quad Y \sim \chi^2(m), \quad Z \perp Y \tag{46}$$

Student's t-distribution

$$X = \frac{Z}{\sqrt{Y}} \sqrt{n} \sim t(n) \tag{47}$$

$$Z \sim N(0,1), \quad Y \sim \chi^2(n), \quad Z \perp Y \tag{48}$$

The Rayleigh distribution:

$$X = \sqrt{Y_1^2 + Y_2^2} \sim Ray(\sigma_y^2), \quad Y_i \sim N_{iid}(0, \sigma_y^2) \tag{49}$$

# Stochastics in Matlab

Sampling from distributions in Matlab

## Stochastic Processes

A stochastic process can be described using a marginal cdf or pdf

$$F_{X_t}(x_t, t) = \Pr\{X_t \leq x_t\} \tag{50}$$

$$f_{X_t}(x_t, t) = \nabla_{x_t} F_{X_t}(x_t, t) \tag{51}$$

or if the different times are related, using joint probabilities

$$F_{X_t, X_s}(x_t, x_s, t, s) = \Pr\{X_t \leq x_t, X_s \leq x_s\} \tag{52}$$

# Properties of Stochastic processes

Mean

$$m_x(t) = \mathbb{E}[x(t)] = \int_{-\infty}^{\infty} z f_{x(t)}(z)\,\mathrm{d}z, \tag{53}$$

Variance

$$P_x(t) = \mathrm{Var}(x(t)) = \mathbb{E}\Big[\big(x(t) - \mathbb{E}[x(t)]\big)\big(x(t) - \mathbb{E}[x(t)]\big)^T\Big], \tag{54}$$

Auto-covariance

$$r_x(t_1, t_2) = \mathrm{Cov}(x(t_1), x(t_2)) = \mathbb{E}\Big[\big(x(t_1) - \mathbb{E}[x(t_1)]\big)\big(x(t_2) - \mathbb{E}[x(t_2)]\big)^T\Big], \tag{55}$$

Note that $r_x(t,t) = P_x(t)$

# Auto-covariance and auto-correlation

Auto-correlation function

$$\rho_x(t_1, t_2) = \frac{r_x(t_1, t_2)}{\sqrt{P_x(t_1)P_x(t_2)}} \tag{56}$$

For stationary processes, only the time difference $\tau = t_1 - t_2$ is relevant

$$r_x(\tau) = \mathrm{Cov}\left[x(t), x(t+\tau)\right] \tag{57a}$$

$$\rho_x(\tau) = \frac{r_x(\tau)}{P_x(\tau)} \tag{57b}$$

# Stationarity

Steady state of deterministic systems: The state does not change in time

Stationary distributions of stochastic systems: The state distribution does not change in time

Strong stationarity

$$f_{x(t_1),\ldots,x(t_n)}(z_1,\ldots,z_n) = f_{x(t_1+\Delta t),\ldots,x(t_n+\Delta t)}(z_1,\ldots,z_n), \tag{58}$$

for any $n \in \mathbb{N}$ and $\Delta t \in \mathbb{R}$.

Weak stationarity: The first two moments (mean and covariance) do not change in time

# Normal process and Markov process

Normal process: Any probability density function $f_{x(t_1),\dots,x(t_n)}(z_1,\dots,z_n)$ is a multivariate normal distribution for any $n \in \mathbb{N}$

Probability density function with mean $\mu$ and covariance $\Sigma$

$$f_Y(y) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(y-\mu)^T\Sigma^{-1}(y-\mu)\right) \tag{59}$$

Markov process: For any $t_1 < t_2 < \cdots < t_n$, the distribution of $x(t_n)$ given $(x(t_1),\dots,x(t_{n-1}))$ is the same as the distribution of $x(t_n)$ given $x(t_{n-1})$

$$\Pr\Big(x(t_n) \leq x \mid x(t_{n-1}),\dots,x(t_1)\Big) = \Pr\Big(x(t_n) \leq x \mid x(t_{n-1})\Big) \tag{60}$$

# Stochastic Vectors

Vector-valued random variables

$$\mathbf{X} = \left[X_1, \ldots, X_n\right]^T \tag{61}$$

$$\text{cdf: } F_{\mathbf{X}}(\mathbf{x}) = \Pr(X_1 \leq x_1, \ldots, X_n \leq x_n, ) \tag{62}$$

$$\text{marginal cdf: } F_{X_1}(x_1) = \Pr(X_1 \leq x_1) \tag{63}$$

1st and 2nd order moments

$$\mathbf{m}_x = \mathbb{E}[\mathbf{X}] = \left[\mathbb{E}[X_1], \ldots, \mathbb{E}[X_n]\right]^T \tag{64}$$

$$P_x = P_x^T = \mathrm{Cov}(\mathbf{X}) = \mathbb{E}[(\mathbf{X} - \mathbf{m}_x)(\mathbf{X} - \mathbf{m}_x)^T] \succeq 0 \tag{65}$$

Positive semi-definiteness ($\succeq$) means that $x^T P_x x \geq 0$ for all $x \neq 0$.

Covariance matrices are diagonalizable, e.g., for $n = 2$

$$P_x = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) \end{bmatrix} \tag{66}$$

# Mathematical properties of Moments: 1st and 2nd

Let $A$ and $m$ be a constant matrix and vector

## Expectations

$$\boxed{\begin{aligned}
\mathbb{E}[\mathbf{X}+m] &= \mathbb{E}[\mathbf{X}]+m && (67)\\
\mathbb{E}[A\mathbf{X}] &= A\mathbb{E}[\mathbf{X}] && (68)\\
\mathbb{E}[\mathbf{X}+\mathbf{Y}] &= \mathbb{E}[\mathbf{X}]+\mathbb{E}[\mathbf{Y}] && (69)\\
\mathbb{E}[\mathbf{X}^T A\mathbf{X}] &= \mathrm{Tr}(A\,\mathrm{Cov}(\mathbf{X}))+\mathbb{E}[\mathbf{X}]^T A\mathbb{E}[\mathbf{X}] && (70)
\end{aligned}}$$

## Covariances

$$\boxed{\begin{aligned}
\mathrm{Cov}(\mathbf{X}) &= \mathbb{E}[\mathbf{X}\mathbf{X}^T]-\mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{X}]^T && (71)\\
\mathrm{Cov}(\mathbf{X}+m) &= \mathrm{Cov}(\mathbf{X}) && (72)\\
\mathrm{Cov}(A\mathbf{X}) &= A\,\mathrm{Cov}(\mathbf{X})A^T && (73)\\
\mathrm{Cov}(\mathbf{X}+\mathbf{Y}) &= \mathrm{Cov}(\mathbf{X})+\mathrm{Cov}(\mathbf{Y})+\mathrm{Cov}(\mathbf{X},\mathbf{Y})+\mathrm{Cov}(\mathbf{X},\mathbf{Y})^T && (74)
\end{aligned}}$$

# Vector Covariance and Variance

Further covariances

$$\boxed{\begin{aligned} \mathrm{Cov}(\mathbf{X},\mathbf{Y}) &= \mathbb{E}[(\mathbf{X}-m_x)(\mathbf{Y}-m_y)^T] && (75)\\ \mathrm{Cov}(\mathbf{X},\mathbf{X}) &= \mathrm{Cov}(\mathbf{X}) = P_x && (76)\\ \mathrm{Cov}(\mathbf{Y},\mathbf{X}) &= \mathrm{Cov}(\mathbf{X},\mathbf{Y})^T && (77)\\ \mathrm{Cov}(A\mathbf{X},\mathbf{Y}) &= A\,\mathrm{Cov}(\mathbf{X},\mathbf{Y}) && (78)\\ \mathrm{Cov}(\mathbf{X},A\mathbf{Y}) &= \mathrm{Cov}(\mathbf{X},\mathbf{Y})A^T && (79)\\ \mathrm{Cov}(\mathbf{X}+\mathbf{V},\mathbf{Y}) &= \mathrm{Cov}(\mathbf{X},\mathbf{Y}) + \mathrm{Cov}(\mathbf{V},\mathbf{Y}) && (80) \end{aligned}}$$

# Auto-regressive models

# The Auto-Regressive (AR) Process

AR($m$) process

$$y_t + \sum_{j=1}^{m} a_j y_{t-j} = e_t, \qquad a_0 = 1 \tag{81}$$

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

# The Auto-Regressive (AR) Process

Shift operator, $q$

$$q^{-1}y_t = y_{t-1}, \tag{82}$$

AR($m$) process (compact notation)

$$A(q^{-1})y_t = e_t \tag{83}$$

Polynomial

$$A(q^{-1}) = 1 + \sum_{j=1}^{m} a_j q^{-j} \tag{84}$$

It is called *auto-regressive* because $y_t$ can be viewed as a regression on past values

$$y_t = e_t - \sum_{j=1}^{m} a_j y_{t-j} \tag{85}$$

# The Auto-Regressive (AR) Process - stability

An AR($m$) process is stable if the roots, $\lambda$, of the characteristic equation

$$a_0 + a_1\lambda^{-1} + \cdots + a_m\lambda^{-m} = 0 \tag{86}$$

lie within the unit circle

Multiply both sides by $\lambda^m$ (assuming $\lambda$ is nonzero)

$$a_0\lambda^m + a_1\lambda^{m-1} + \cdots + a_m = 0 \tag{87}$$

Hint: If the Matlab vector `A` contains the elements $a_0$, $a_1$, ..., $a_m$ (in that order), you can use the command `roots(A)` to find the poles (double-check with the documentation)

# The Auto-Regressive (AR) Process – Mean

Mean

$$A(q^{-1})\mathbb{E}[y_t] = \mathbb{E}[e_t] = 0 \tag{88}$$

Stationary mean, $\mathbb{E}[y_t] = \bar{y}$ for all $t$

$$A(q^{-1})\mathbb{E}[y_t] = A(q^{-1})\bar{y} = A(1)\bar{y} = 0 \tag{89}$$

$A(q^{-1})$ applied to a constant (e.g., $\bar{y}$) is just the sum of the coefficients

$$A(1) = \sum_{j=0}^{m} a_j \tag{90}$$

If $A(1) \neq 0$

$$\bar{y} = 0 \tag{91}$$

# The Auto-Regressive (AR) Process – Auto-covariance

Auto-covariance function of an AR($m$) process

$$\gamma(i) = \mathrm{Cov}(y_t, y_{t-i}) \tag{92}$$

Auto-covariance

$$\gamma(i) + \sum_{j=1}^{m} a_j \gamma(i-j) = 0, \qquad i > 0 \tag{93}$$

Initial condition

$$\gamma(0) + \sum_{j=1}^{m} a_j \gamma(j) = \sigma_e^2 \tag{94}$$

Symmetry of auto-covariance functions: $\gamma(i) = \gamma(-i)$

# The Auto-Regressive Process with Exogenous inputs (ARX)

ARX process

$$y_t + \sum_{j=1}^{m} a_j y_{t-j} = \sum_{j=0}^{n} b_j u_{t-j} + e_t, \qquad a_0 = 1 \tag{95}$$

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

ARX process (using polynomials)

$$A(q^{-1})y_t = B(q^{-1})u_t + e_t \tag{96}$$

# The Auto-Regressive Process with Exogenous inputs (ARX)

Mean

$$A(q^{-1})\mathbb{E}[y_t] = B(q^{-1})\mathbb{E}[u_t] + \mathbb{E}[e_t] \tag{97}$$

Stationary mean, $\mathbb{E}[y_t] = \bar{y}$ and $\mathbb{E}[u_t] = u_t = \bar{u}$ for all $t$

$$A(q^{-1})\mathbb{E}[y_t] = A(1)\bar{y} = B(q^{-1})\mathbb{E}[u_t] = B(1)\bar{u} \tag{98}$$

If $A(1) \neq 0$

$$\bar{y} = \frac{B(1)}{A(1)}\bar{u} \tag{99}$$

# The Moving-Average (MA) Process

MA($n$) process

$$y_t = e_t + \sum_{j=1}^{n} c_j e_{t-j}, \qquad c_0 = 1 \qquad (100)$$

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

# The Moving-Average (MA) Process

MA($n$) process (compact notation)

$$y_t = C(q^{-1})e_t, \tag{101}$$

Polynomial

$$C(q^{-1}) = 1 + \sum_{j=1}^{n} c_j q^{-j} \tag{102}$$

# 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

$$\gamma(i) = \begin{cases} \sigma^2\Big(c_i + c_1 c_{i+1} + \cdots + c_{n-i}c_n\Big), & |i| = 0, \ldots, n, \\ 0, & |i| > n \end{cases} \tag{103}$$

Variance (constant)

$$\sigma_y^2 = \gamma(0) = \sigma^2 \left(1 + \sum_{j=1}^{n} c_j^2\right) \tag{104}$$

# The ARMA Process

ARMA($m$,$n$) process

$$y_t + \sum_{j=1}^{m} a_j y_{t-j} = e_t + \sum_{j=1}^{n} c_j e_{t-j} \tag{105}$$

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

# The ARMA Process

ARMA($m$,$n$) process (compact notation)

$$A(q^{-1})y_t = C(q^{-1})e_t \tag{106}$$

Polynomials

$$A(q^{-1}) = 1 + \sum_{j=1}^{m} a_j q^{-j}, \qquad C(q^{-1}) = 1 + \sum_{j=1}^{n} c_j q^{-j} \tag{107}$$

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

# ARMA – Stationary mean

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

$$\mathbb{E}\left[\sum_{j=0}^{m} a_j y_{t-j}\right] = \mathbb{E}\left[\sum_{j=0}^{n} c_j e_{t-j}\right], \tag{108}$$

$$\sum_{j=0}^{m} a_j \mathbb{E}[y_{t-j}] = \sum_{j=0}^{n} c_j \mathbb{E}[e_{t-j}] = 0, \tag{109}$$

$$\sum_{j=0}^{m} a_j \bar{y} = \left(\sum_{j=0}^{m} a_j\right) \bar{y} = 0 \tag{110}$$

If $A(1) = \sum_{j=0}^{m} a_j \neq 0$

$$\bar{y} = 0 \tag{111}$$

# ARMA – Stationary cross-covariance

Covariance (lag 0)

$$\mathrm{Cov}\left(\sum_{j=0}^{m} a_j y_{t-j}, e_t\right) = \mathrm{Cov}\left(\sum_{j=0}^{n} c_j e_{t-j}, e_t\right) = c_0 \,\mathrm{Cov}(e_t, e_t) = c_0 \sigma^2 \tag{112}$$

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

$$\mathrm{Cov}\left(\sum_{j=0}^{m} a_j y_{t-j}, e_t\right) = a_0 \,\mathrm{Cov}(y_t, e_t) = a_0 \gamma_{ye}(0) \tag{113}$$

Covariance (lag 0)

$$\gamma_{ye}(0) = \frac{c_0}{a_0} \sigma_\varepsilon^2 \tag{114}$$

# ARMA – Stationary cross-covariance

Covariance (lag 1)

$$\mathrm{Cov}\left(\sum_{j=0}^{m} a_j y_{t-j}, e_{t-1}\right) = \mathrm{Cov}\left(\sum_{j=0}^{n} c_j e_{t-j}, e_{t-1}\right) \tag{115}$$

$$= c_1 \,\mathrm{Cov}(e_{t-1}, e_{t-1}) = c_1 \sigma^2 \tag{116}$$

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

$$\mathrm{Cov}\left(\sum_{j=0}^{m} a_j y_{t-j}, e_{t-1}\right) = a_0 \,\mathrm{Cov}(y_t, e_{t-1}) + a_1 \,\mathrm{Cov}(y_{t-1}, e_{t-1}) \tag{117}$$

$$= a_0 \gamma_{ye}(1) + a_1 \gamma_{ye}(0) \tag{118}$$

Covariance (lag 1)

$$\gamma_{ye}(1) = \frac{c_1}{a_0}\sigma^2 - \frac{a_1}{a_0}\gamma_{ye}(0) \tag{119}$$

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

# ARMA – Stationary cross-covariance

Covariance

$$\mathrm{Cov}\left(\sum_{j=0}^{m} a_j y_{t-j}, y_{t-i}\right) = \mathrm{Cov}\left(\sum_{j=0}^{n} c_j e_{t-j}, y_{t-i}\right), \quad i = 0, \dots, m \quad (120)$$

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

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{m-1} & a_m \\ a_1 & a_0 + a_2 & a_3 & \cdots & a_m & 0 \\ a_2 & a_1 + a_3 & a_0 + a_4 & \cdots & 0 & 0 \\ a_3 & a_2 + a_4 & a_1 + a_5 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{m-1} & a_{m-2} + a_m & a_{m-3} & \cdots & a_0 & 0 \\ a_m & a_{m-1} & a_{m-2} & \cdots & a_1 & a_0 \end{bmatrix} \begin{bmatrix} \gamma(0) \\ \gamma(1) \\ \gamma(2) \\ \gamma(3) \\ \vdots \\ \gamma(m-1) \\ \gamma(m) \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{m-1} \\ r_m \end{bmatrix} \quad (121)$$

$$r_i = \sum_{j=i}^{n} c_j \gamma_{ye}(j-i)$$

Custom Matlab routine: trfvar2.m

# ARMAX model with constant disturbance

ARMAX

$$A(q^{-1})y_t = q^{-k}B(q^{-1})u_t + C(q^{-1})e_t + d \tag{122}$$

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

$$\mathbb{E}\left[\sum_{j=0}^{n_a} a_j y_{t-j}\right] = \mathbb{E}\left[\sum_{j=0}^{n_b} b_j u_{t-k-j} + \sum_{j=0}^{n_c} c_j e_{t-j} + d\right], \tag{123}$$

$$\sum_{j=0}^{n_a} a_j \mathbb{E}[y_{t-j}] = \sum_{j=0}^{n_b} b_j u_{t-k-j} + \sum_{j=0}^{n_c} c_j \mathbb{E}[e_{t-j}] + d = \sum_{j=0}^{n_b} b_j u_{t-k-j} + d,$$

$$\sum_{j=0}^{n_a} a_j \bar{y} = \left(\sum_{j=0}^{n_a} a_j\right)\bar{y} = \sum_{j=0}^{n_b} b_j u_{t-k-j} + d = \left(\sum_{j=0}^{n_b} b_j\right)\bar{u} + d \tag{124}$$

If $A(1) = \sum_{j=0}^{n_a} a_j \neq 0$

$$\bar{y} = \frac{B(1)}{A(1)}\bar{u} + \frac{1}{A(1)}d \tag{125}$$

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 $j = 0, \ldots, n_c$

$$\begin{aligned} y_t = b_0 u_{t-k} + \cdots + b_{n_b} u_{t-k-n_b} + c_0 e_t + \cdots + c_{n_c} e_{t-n_c} \\ - a_1 y_{t-1} - \cdots - a_{n_a} y_{t-n_a} + d \end{aligned} \tag{126}$$

## 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' + d;
```

# Questions?