Chapter 3

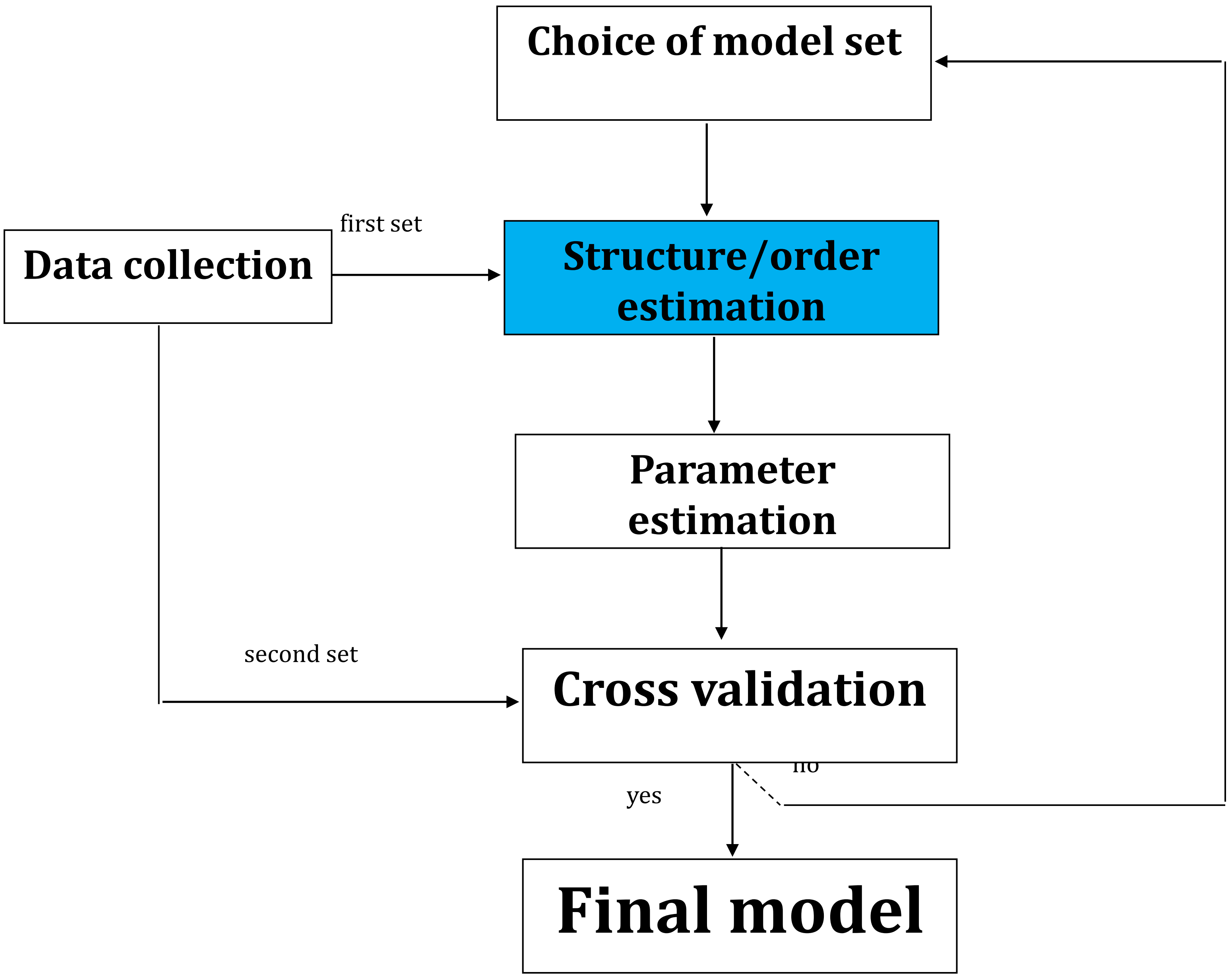
Model Structure Selection

Introduction

- Parametric models describe systems in terms of **difference or differential equations** depending on whether a system is represented by a **discrete or continuous model**.
- Compare to nonparametric models, **parametric models might provide more accurate estimation if users have prior knowledge about system dynamics** to determine model orders, time delays and so on.

Steps in the identification process

- System identification is a very complex process.
- Procedure for system ID as shown in the flow chart,


Choice of model set
first set
Data collection
Structure/order estimation
Parameter estimation
second set
Cross validation
no
yes
Final model

Choice of the model set

- A primary and very important choice is that of the class of the models to be considered.
- A model is a mathematical description of the studied system, and it is possible to divide a set of models into different categories using specific criteria.

Criteria of model

- **Parametric or nonparametric models**
- **Linear and nonlinear models**
- **White box or black box model**

Parametric or nonparametric models

- In a parametric model, the system is described using a limited number of characteristic quantities called the **parameters** of the model, while in a nonparametric model the system is characterized by **measurement of a system function** at a large number of points.

Linear & nonlinear models

- In real life almost every system is **nonlinear**. Thus, nonlinear mathematical models are the most desired ones.
- This approach needs deep understanding of the underlying mechanisms that govern the system behaviour. In practice, this mechanistic approach has a couple of disadvantages.

Linear and nonlinear models

- The development time of the model is long, the values of some parameters of the model may be uncertain due to lack of information and due to fact that they must be determined empirically, and after the refinements the model may become too complex to run at acceptable speed on a computer.

Linear and nonlinear models

- Linear models have had a great success and their use is justified by the following facts:
  - linearization around the equilibrium yield mathematically tractable linear models
  - the output of system can be computed for any arbitrary input
  - most nonlinear systems could be approximated satisfactorily in their normal ranges of operation.

Categories of parametric models

- General linear polynomial*
- Transfer function*
- Zero-pole gain*
- State space
- * = polynomial models

General linear polynomial model

- This model apply only for **discrete systems**
- System can be describe using the following equation

$$y(k) = z^{-n} G(z^{-1}, \theta) u(k) + H(z^{-1}, \theta) e(k)$$

- u(k) and y(k) are the input and output of the system
- $G(z^{-1},\theta)$ is the transfer function of the deterministic part of the system
- $H(z^{-1},\theta)$ is the transfer function of the stochastic part of the system

General linear polynomial model

- The **deterministic transfer function** specifies **the relationship between the output and the input signal**.
- The **stochastic transfer function** specifies **how the random disturbance affects the output signal**.
- Often the deterministic and stochastic parts of a system are referred to as system dynamics and stochastic dynamics respectively

General linear polynomial model

- The term $\mathbf{z^{-1}}$ is **the backward shift operator**, which is defined by the following equations,

$$z^{-1}x(k) = x(k-1)$$

$$z^{-2}x(k) = x(k-2)$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$z^{-n}x(k) = x(k-n)$$

- z^{-n} defines the number of delay samples between the input and the output

General linear polynomial model

- $G(z^{-1},\theta)$ and $H(z^{-1},\theta)$ are rational polynomials as defined by the following equations :

$$G(z^{-1},\theta) = \frac{B(z,\theta)}{A(z,\theta)F(z,\theta)}$$

$$H(z^{-1},\theta) = \frac{C(z,\theta)}{A(z,\theta)D(z,\theta)}$$

General linear polynomial model

- The **vector θ** is **the set of model parameters.**
- Equations in the following slides do not display θ to make the equation easier to read.

General linear polynomial model

- The following equations define A(z), B(z), C(z), D(z) and F(z).

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{k_a} z^{-k_a}$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{k_b - 1} z^{-(k_b - 1)}$$

$$C(z) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{k_c} z^{-k_c}$$

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + \ldots + a_{k_d} z^{-k_d}$$

$$F(z) = 1 + f_1 z^{-1} + f_2 z^{-2} + \ldots + f_{k_f} z^{-k_f}$$

- where k_a, k_b, k_c, k_d and k_f are the model orders

General linear polynomial model

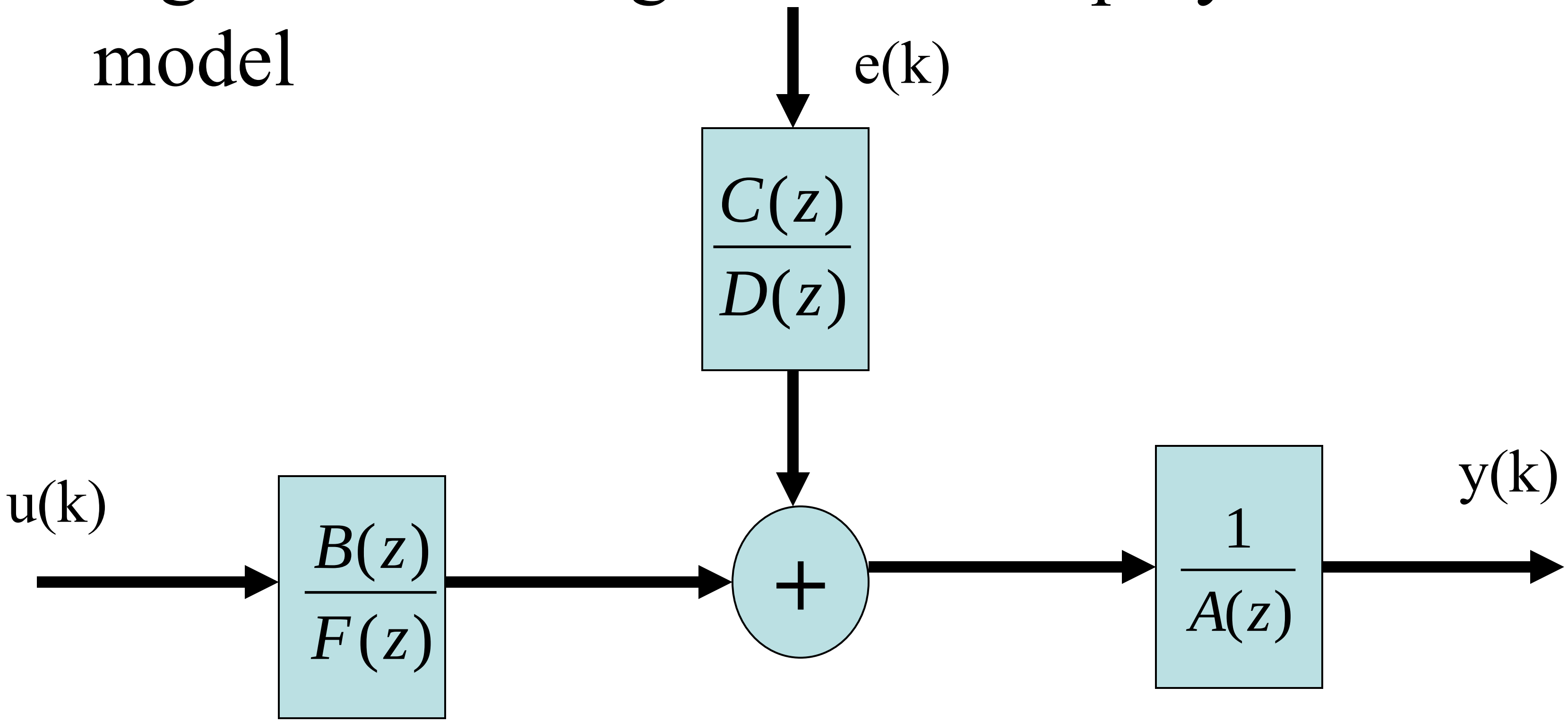
- The following equation describes a general linear polynomial model

$$A(z)y(k) = \frac{z^{-n}B(z)}{F(z)}u(k) + \frac{C(z)}{D(z)}e(k)$$

$$A(z)y(k) = \frac{B(z)}{F(z)}u(k-n) + \frac{C(z)}{D(z)}e(k)$$

General linear polynomial model

- Signal flow of a general linear polynomial model



General linear polynomial model

- A general linear polynomial model provides flexibility for both system dynamics and stochastic dynamics.
- Setting one or more of A(z), C(z), D(z) and F(z) **equal to 1** create simpler models such as **autoregressive with exogenous terms (ARX), autoregressive moving average with exogenous terms (ARMAX), output error and Box Jenkins models** which are commonly use in real world application.

ARX Model

- When **C(z), D(z) and F(z) equal 1**, the general linear polynomial model reduces to an **ARX model**.
- The following equation describes an ARX model

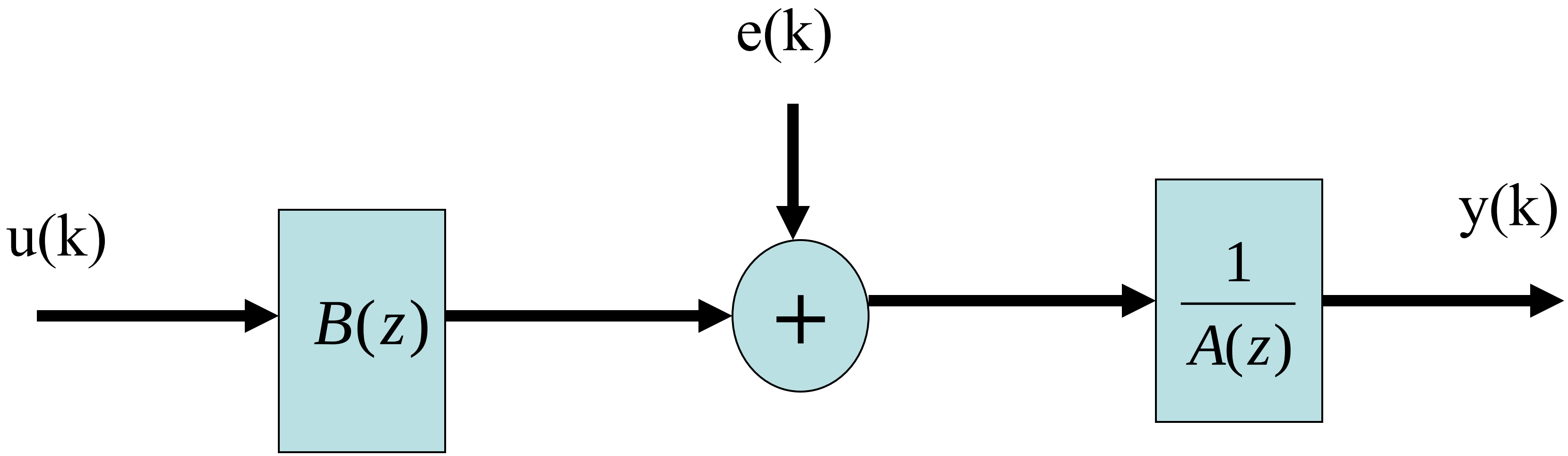
$$A(z)y(k) = z^{-n}B(z)u(k) + e(k)$$

$$A(z)y(k) = B(z)u(k-n) + e(k)$$

- The backward shift operator makes
- $\mathbf{z^{-n}u(k)=u(k-n)}$

ARX Model

- **Signal flow graph of an ARX Model**



ARX Model

- The ARX model is the simplest model incorporating the stimulus signal.
- However, the ARX model captures some of the stochastic dynamics as part of the system dynamics.
- In this model, the transfer function of the deterministic part $G(z^{-1},\theta)$ of the system and the transfer function of the stochastic part $H(z^{-1},\theta)$ of the system **have the same set poles.**

ARX Model

- This coupling can be unrealistic.
- The system dynamics and stochastic dynamics of a system do not share the same set of poles all the time.
- This disadvantage can be reduce if the signal to noise ratio is high.

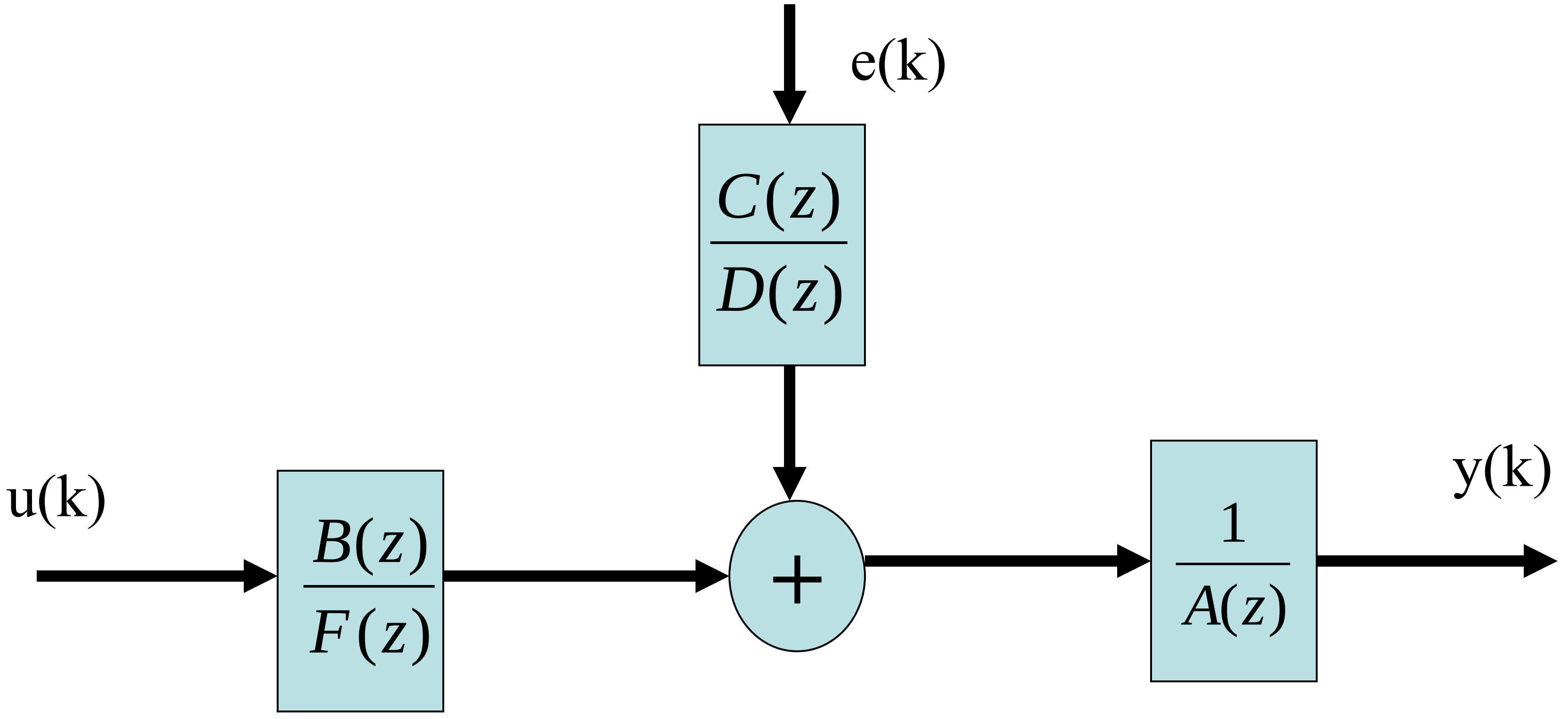
ARX Model

- When disturbance e(k) of a system is not white noise, the coupling between the deterministic and stochastic dynamic can bias the estimation of the ARX model.
- The model order can be set higher than the actual model order to minimize the estimation error especially when the signal to noise ratio is low.

ARX Model

- However increasing the model order can change some dynamic characteristics of the model such as the stability of the model.
- The identification method for the ARX model is the **least squares method**.
- The least square method is the most efficient polynomial estimation method because this method solves linear regression equations in analytic form.

General linear polynomial model



ARMAX Model

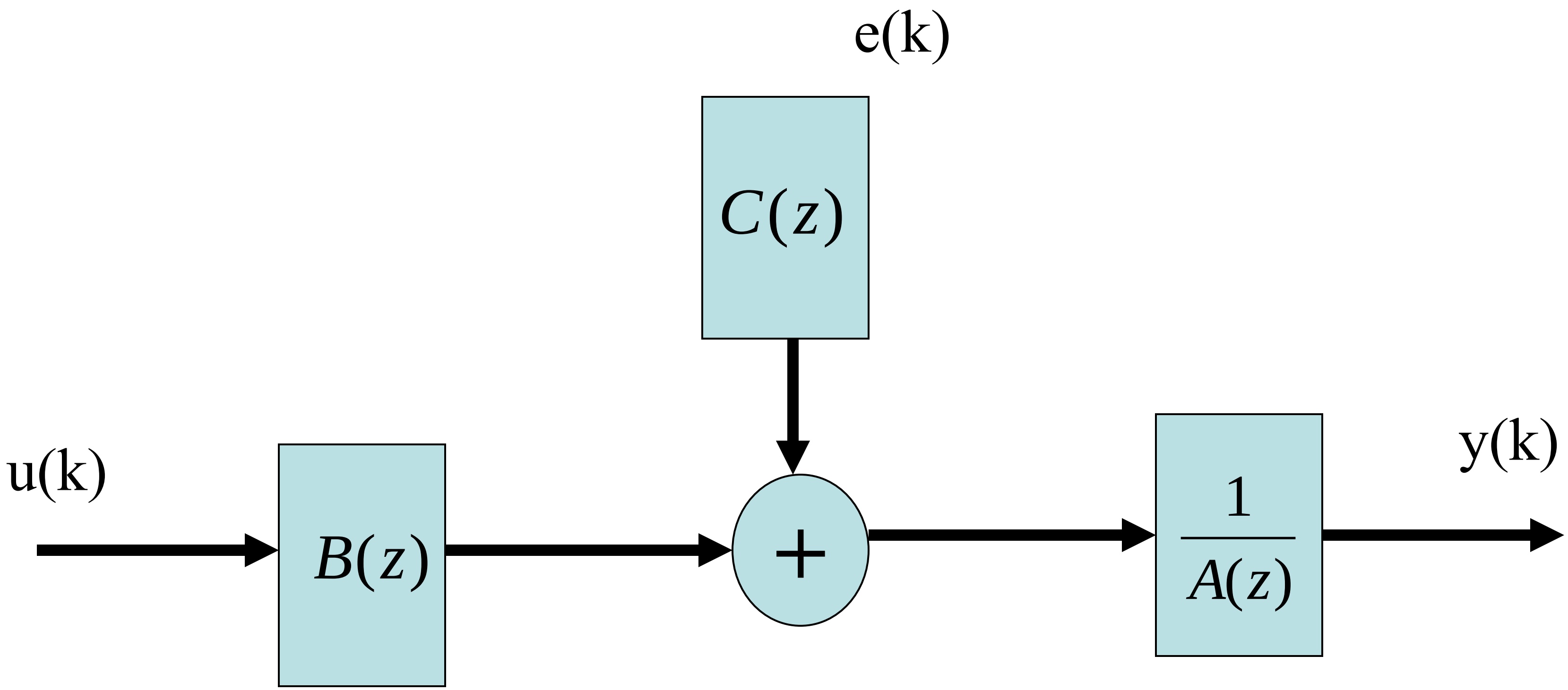
- When **D(z) and F(z) equal 1**, the general linear polynomial model reduces to the **ARMAX model**.
- The following equation describes an ARMAX model.

$$A(z)y(k) = z^{-n}B(z)u(k) + C(z)e(k)$$

$$A(z)y(k) = B(z)u(k-n) + C(z)e(k)$$

ARMAX Model

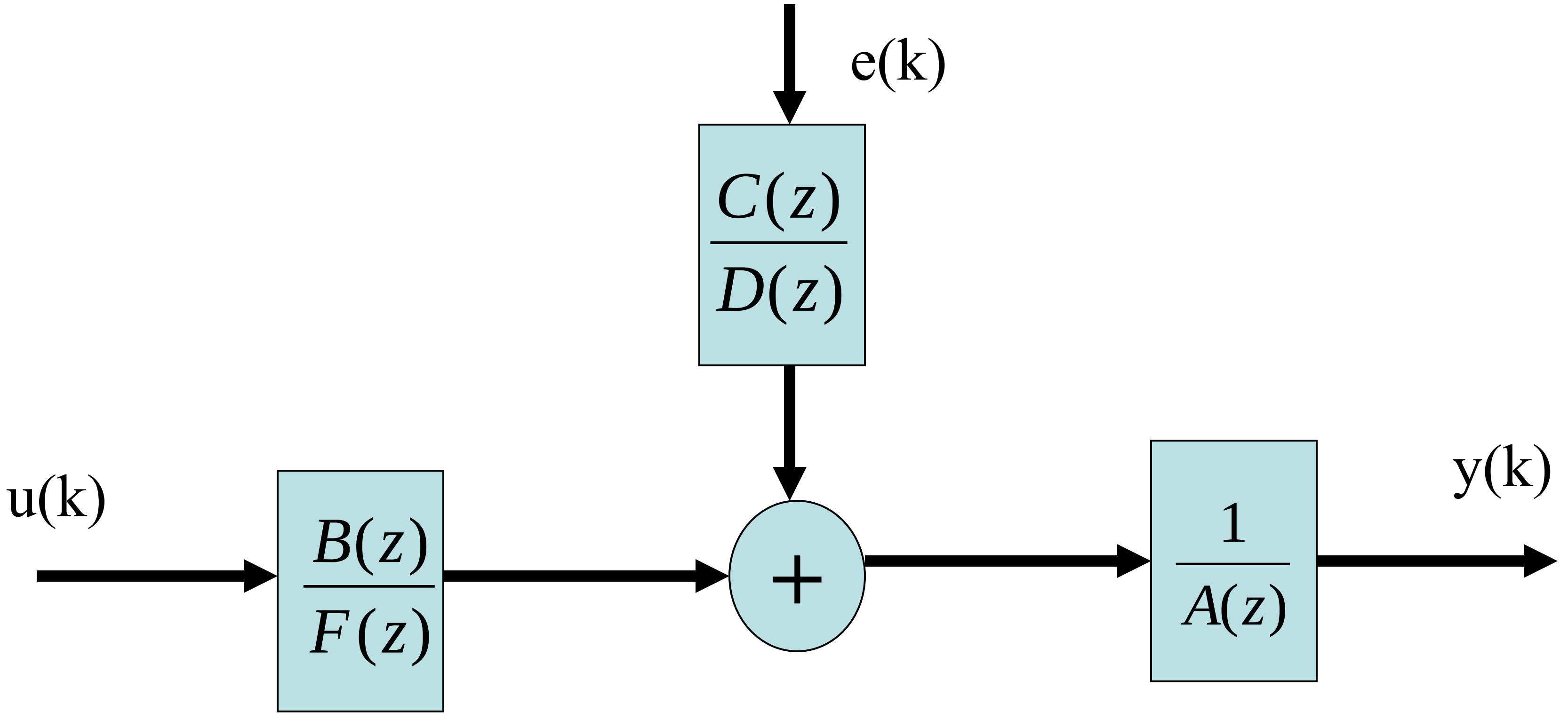
- **Signal flow of an ARMAX Model**



ARMAX Model

- Unlike the ARX model, the system structure of an ARMAX model **includes the stochastic dynamics.**
- ARMAX models are useful when **dominating disturbances exist that enter early in the process such as the input.**
- The ARMAX model has **more flexibility** than the ARX model in handling models that **contain disturbances**.
- The identification method of the ARMAX model is the **prediction error method**.

General linear polynomial model



Output Error Model

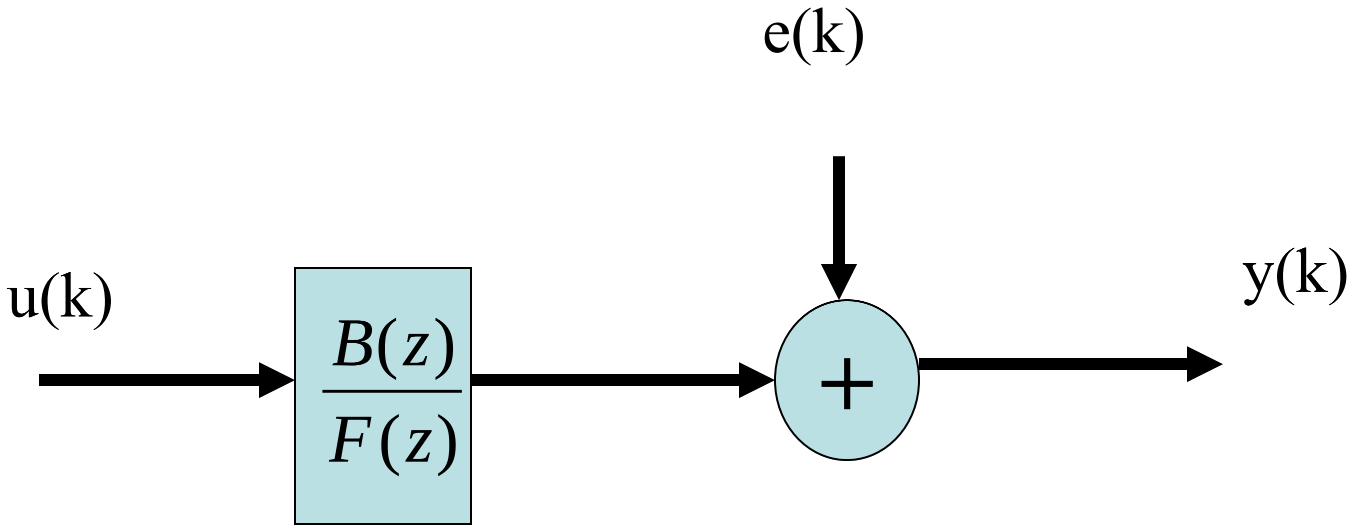
- When **A(z), C(z) and D(z) equal 1**, the general linear polynomial model reduces to the **output error model**.
- The following equation describes an output error model

$$y(k) = \frac{z^{-n} B(z)}{F(z)} u(k) + e(k)$$

$$y(k) = \frac{B(z)}{F(z)} u(k-n) + e(k)$$

Output Error Model

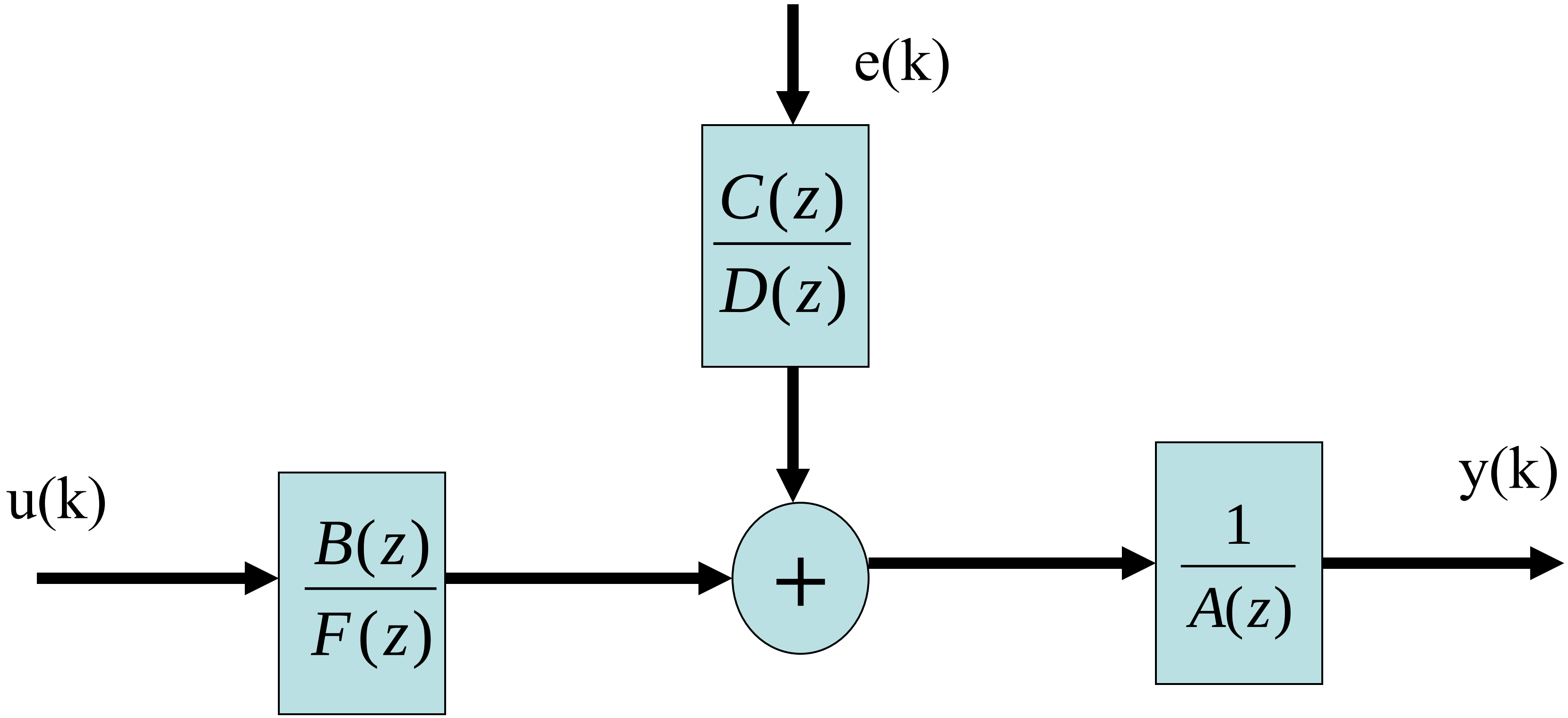
- **Signal flow of an output error model**



Output Error Model

- The output error model **describes the system dynamics separately from the stochastic dynamics.**
- The output error model **does not use any parameters for simulating the disturbance characteristics.**
- The identification method of the output error model is the **prediction error method** which is the same as that of the ARMAX model.

General linear polynomial model



Box and Jenkins Model

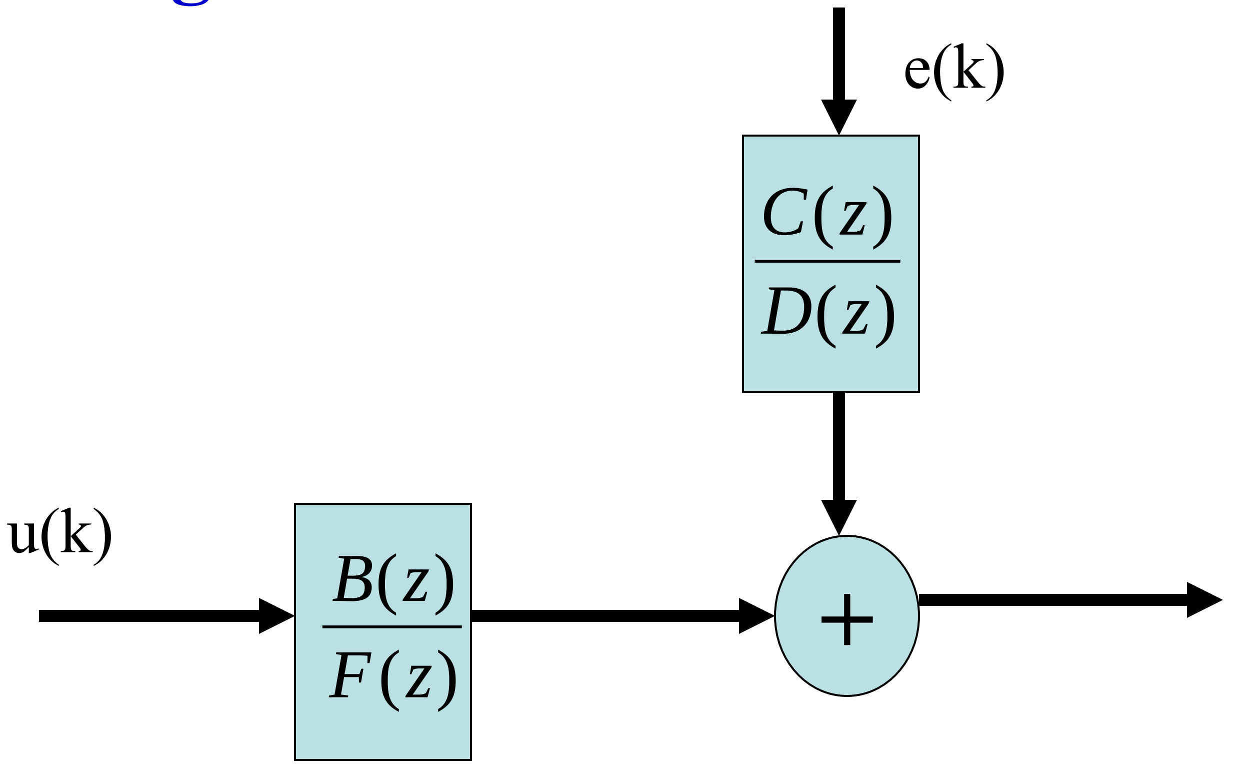
- When **A(z) equals 1**, the general linear polynomial model reduces to the **Box and Jenkins model.**
- The following equation describes a Box Jenkins model

$$y(k) = \frac{z^{-n}B(z)}{F(z)}u(k) + \frac{C(z)}{D(z)}e(k)$$

$$y(k) = \frac{B(z)}{F(z)}u(k-n) + \frac{C(z)}{D(z)}e(k)$$

Box and Jenkins Model

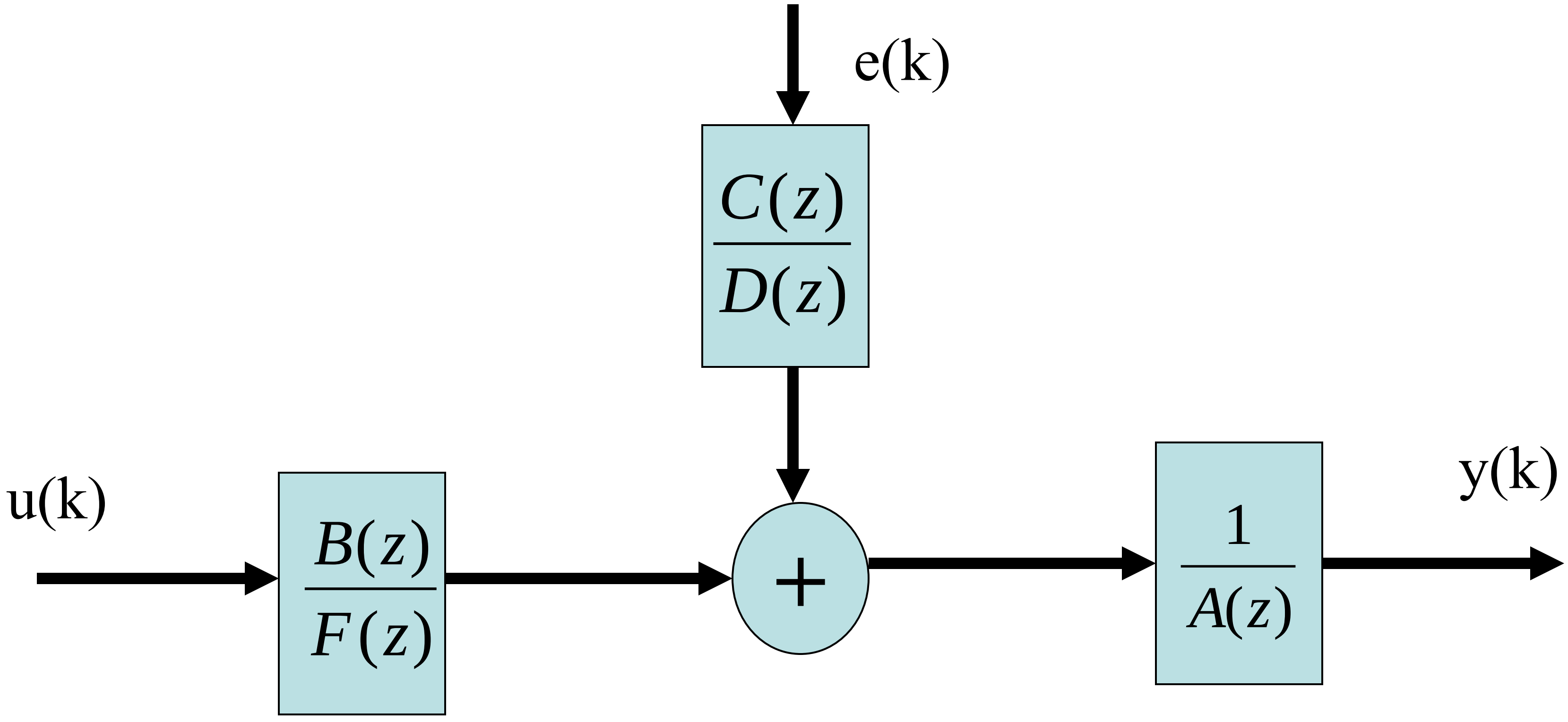
- **Signal flow of a Box and Jenkins Model**



Box and Jenkins Model

- The Box and Jenkins model **provides a complete model of a system.**
- The Box and Jenkins model represents **disturbance properties separately from system dynamics.**
- This model is **useful when you have disturbances that enter late in the process such as measurement noise on the output**.
- The identification method of the Box and Jenkins model is the **prediction error method** which is the same that of the ARMAX model.

General linear polynomial model

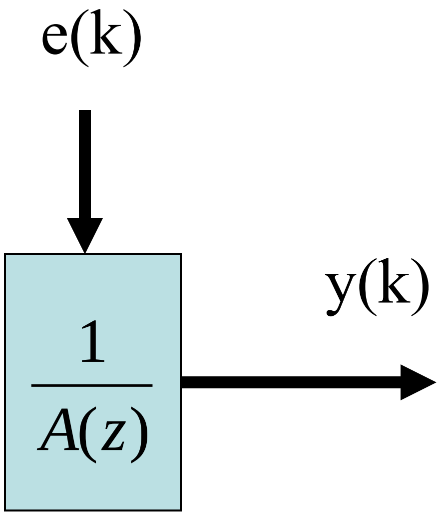


AR Model

- When **C(z), D(z) and F(z) equal 1 and B(z) equal 0**, the general linear polynomial model reduces to the **AR model.**
- The following equation describes and AR model

AR Model

- **Signal flow of an AR Model**



AR model

- The AR model **does not include the dynamics between the input and output**.
- Therefore the AR model is more **suitable for representing signals rather than a system** because a system generally has an input and an output
- **Time series analysis methods**, such as **power spectrum envelope estimation, pre-whitening and linear prediction coding** commonly use AR model

Transfer Function Model

- Transfer function can be define either a continuous system or a discrete system using the following equations

$$y(t) = G(s)u(t) + e(t)$$

$$y(k) = G(z)u(k) + e(k)$$

Transfer Function Model

- Parameters of the model are the numerator and denominator coefficients of the transfer function.
- The transfer function provides the a mathematical representation of the relationship between one input and one output.

Transfer Function Model

- The following equations define the continuous and discrete transfer functions where the numerator and denominator are polynomials.
- **Continuous transfer function model**

$$G(s) = \frac{b_0 + b_1 s + \ldots + b_{m-1} s^{m-1} + b_m s^m}{a_0 + a_1 s + \ldots + a_{n-1} s^{n-1} + a_n s^n}$$

- **Discrete transfer function model**

$$G(z) = \frac{b_0 + b_1 z + \ldots + b_{m-1} z^{m-1} + b_m z^m}{a_0 + a_1 z + \ldots + a_{n-1} z^{n-1} + a_n z^n}$$

Transfer Function Model

- Transfer function models **describes only the deterministic part of the system**.
- For stochastic control, general linear polynomial models commonly are used because these models separately describe the deterministic and stochastic part of a system.
- However in classical control engineering the deterministic part of the system is more important than the stochastic part of the system.
- The relationship between input and output signals of the transfer function model describe the deterministic part of the system

Zero-Pole Gain Model

- The transfer function model equation that **show the locations of the zeroes and poles of the dynamic system**
- The following equation represent the continuous and discrete zero pole gain model

$$G(s) = \frac{K(z - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)}$$

$$G(z) = \frac{K(z - z_1)(z - z_2)\ldots(z - z_m)}{(z - p_1)(z - p_2)\ldots(s - p_n)}$$

- Where K is the gain, z_i are he zeroes and p_j are the poles

State Space Model

- The state space model **describes a system using difference or differential equations with an auxiliary state vector**.
- The following equations describe a discrete state space model.

$$x(k+1) = Ax(k) + Bu(k) + Ke(k)$$
$$y(k) = Cx(k) + Du(k) + e(k)$$

State Space Model

- The following equations describe a continuous state space model.

$$\dot{x} = Ax(t) + Bu(t) + Ke(t)$$

$$y = Cx(t) + Du(t) + e(t)$$

State Space Model

- Where,

x = state vector
k = model sampling time x discrete time step
t = time
A = system matrix
B = input matrix
C = output matrix
D = transmission matrix
K = Kalman gain

State Space Model

- The dimension of the state vector x is the important factor for the state space model
- The state space transfer matrices A, B, C and D often reflect physical characteristics of a system.
- The state space model is the most convenient model in **describing multivariable system.**
- State space models often are preferable to polynomial models especially n modern control applications that focus on multivariable system.

State Space Model

- To estimate discrete state space models, use 2 type of identification methods
  - **Deterministic stochastic subspace method**
  - **Realization method**

State Space Model

- Deterministic stochastic subspace method
  - Uses principal component analysis to estimate parameters
  - This method uses both stimulus and response signal to estimate ss models

State Space Model

- Realization method
  - Uses the impulse response to estimate only the deterministic ss model
  - This method does not include stochastic parts of the system in the model structure

State Space Model

- Therefore the difference between these two methods is that **the deterministic stochastic subspace method includes noise in the model structure whereas the realization method does not.**

Polynomial models versus state space models

- Selecting the correct model type and model order is crucial for successfully estimating a parametric model.
- In general state space models provide a more complete representation of the system especially for multiple input multiple output (MIMO) systems than polynomial model because state space models are similar to first principle models that can provide more degree of freedom in describing MIMO systems.

Polynomial models versus state space models

- The identification procedure for state space models does not involve nonlinear optimization so the estimation reaches a solution regardless of the initial guess.
- Moreover the parameter settings for the state space model are simpler.
- Select the order or the number of states of the model.
- The order can come from prior knowledge of the system.

Polynomial models versus state space models

- The order can also be determine by analyzing the singular values of the information matrix.
- However the states that the state space identification procedure identifies might not reflect the physical characteristic of a system accurately.
- Using similarity transformation can identify equivalent models with states that better represent the system.

Polynomial models versus state space models

- Similarity transformation can transform the states without misrepresenting the input output behavior of the system.
- When model order is high, state space model are preferable to polynomial models.
- Polynomial models with high order might encounter numerical problems in computation

Nonlinear autoregressive exogenous model: NARX model

- In time series modeling, a **nonlinear autoregressive exogenous model** (NARX) is a ***nonlinear autoregressive model*** which has ***exogenous*** inputs.
- This means that the model relates the current value of a time series to both:
  - past values of the same series
  - current and past values of the driving (exogenous) series
  - an "error" term

Nonlinear autoregressive exogenous model: NARX model

- Such a model can be stated algebraically as:

$$y_t = F(y_{t-1}, y_{t-2}, y_{t-3}, \ldots\ldots\ldots u_t, u_{t-1}, u_{t-2}, u_{t-3}, \ldots\ldots) + \varepsilon_t$$

- where
  - y = variable of interest
  - u = externally determined variable, u helps to predict y
  - ε = error term or noise

The function *F* is some nonlinear function, such as a polynomial. *F* can be a neural network, a wavelet network, a sigmoid network and others.

Nonlinear autoregressive moving average exogenous model: NARMAX model

- The **n**onlinear **a**uto**r**egressive **m**oving **a**verage model with e**x**ogenous inputs (NARMAX model) can represent a wide class of nonlinear systems, and is defined as:

$$y(k) = F = [y(k-1), y(k-2), \ldots y(k-n_y), u(k-d), u(k-d-1), \ldots, u(k-d-n_y),$$
$$e(k-1), e(k-2), \ldots, e(k-n_e)] + e(k)$$

where $y(k)$, $u(k)$ and $e(k)$ are the system output, input, and noise sequences respectively;

where ny, nu and ne are the maximum lags for the system output, input and noise

F[*] is some nonlinear function, d is a time delay typically set to $d = 1$