ELEC4410

Control System Design

Lecture 2: Mathematical Description of Systems

School of Electrical Engineering and Computer Science
The University of Newcastle

Outline

- A Taxonomy of Systems

Outline

- A Taxonomy of Systems
- Linear Systems

Outline

- A Taxonomy of Systems
- Linear Systems
- Linear Time-Invariant Systems

Outline

- A Taxonomy of Systems
- Linear Systems
- Linear Time-Invariant Systems
- Linearisation

Outline

- A Taxonomy of Systems
- Linear Systems
- Linear Time-Invariant Systems
- Linearisation
- Discrete-Time Systems

Outline

- A Taxonomy of Systems
- Linear Systems
- Linear Time-Invariant Systems
- Linearisation
- Discrete-Time Systems
- A Few General Facts to Remember

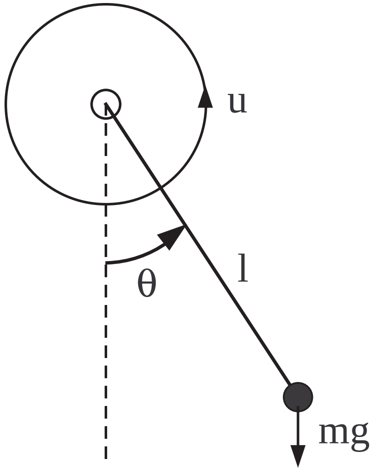
Outline

- A Taxonomy of Systems
- Linear Systems
- Linear Time-Invariant Systems
- Linearisation
- Discrete-Time Systems
- A Few General Facts to Remember

Reference: Linear System Theory and Design, Chen.

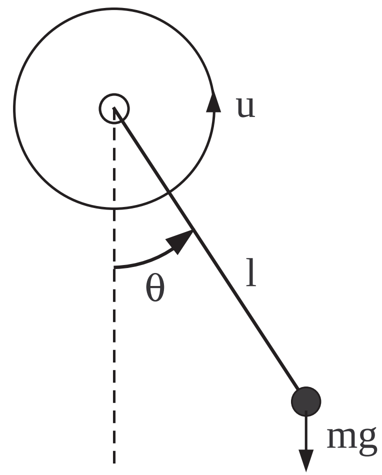
A Taxonomy of Systems

- Consider a simple problem in robotics, i.e. control of the position of a robot arm using a motor located at the arm joint.



A Taxonomy of Systems

- Consider a simple problem in robotics, i.e. control of the position of a robot arm using a motor located at the arm joint.



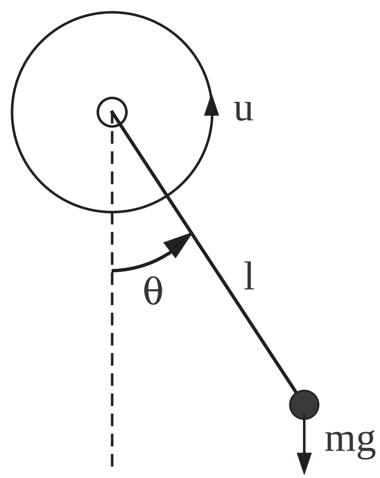
- Mathematically, this system is nothing else than a pendulum controlled by torque.

A Taxonomy of Systems

- Assume:
  - friction at the joint is negligible,
  - the arm is rigid, and
  - all the mass of the arm is concentrated on its free end,

then angle with respect to the vertical θ is given by the differential equation

$$ml^2\ddot{\theta}(t) + mgl\sin\theta(t) = u(t)\,.$$



A Taxonomy of Systems

- The single robot arm model given by the differential equation

$$ml^2\ddot{\theta}(t) + mgl\,\sin\theta(t) = u(t)\,.$$

is an example of a system that is:

- dynamic
- causal
- finite-dimensional
- continuous-time
- nonlinear
- time-invariant

A Taxonomy of Systems

Dynamic /Static?

- *Dynamic* means that the variables θ and $\dot{\theta} \doteq d\theta(t)/dt$, which define the *state* of the arm at a given instant of time t, have a non *instantaneous* dependency on the control torque u. A dynamic system is said to possess memory, i.e. it output depends also on previous inputs.

A Taxonomy of Systems

Dynamic /Static?

- *Dynamic* means that the variables θ and $\dot{\theta} \doteq d\theta(t)/dt$, which define the *state* of the arm at a given instant of time t, have a non *instantaneous* dependency on the control torque u. A dynamic system is said to possess memory, i.e. it output depends also on previous inputs.

- A system that is *not* dynamic is called *static*. In a static system the output has an instantaneous dependency on the evolution of the input. Static systems are also called *memoryless*.

A Taxonomy of Systems

Causal?

- *Causal* means that the output of the system at a given instant of time only depends on present and past values of the input, and not on *future values*.

A Taxonomy of Systems

Causal?

- *Causal* means that the output of the system at a given instant of time only depends on present and past values of the input, and not on *future values*.
- In a causal system the output cannot anticipate or predict future values of the input.

A Taxonomy of Systems

Causal?

- *Causal* means that the output of the system at a given instant of time only depends on present and past values of the input, and not on *future values*.
- In a causal system the output cannot anticipate or predict future values of the input.
- All real physical systems are causal.

A Taxonomy of Systems

Causal?

- *Causal* means that the output of the system at a given instant of time only depends on present and past values of the input, and not on *future values*.
- In a causal system the output cannot anticipate or predict future values of the input.
- All real physical systems are causal.
- The current output of a causal dynamic system always depends on past values of the input. But how far back in time do these past values still have an effect on the output?

A Taxonomy of Systems

Causal?

- *Causal* means that the output of the system at a given instant of time only depends on present and past values of the input, and not on *future values*.
- In a causal system the output cannot anticipate or predict future values of the input.
- All real physical systems are causal.
- The current output of a causal dynamic system always depends on past values of the input. But how far back in time do these past values still have an effect on the output?
  - Strictly, we would need to go back in time up to $t = -\infty$, which is not very practical. This difficulty is resolved with the concept of *state*.

A Taxonomy of Systems

State?

- The *state* $x(t_0)$ of a system at the time instant t_0 is the information that together with the input $u(t)$ for $t \geq t_0$ univocally determines the output $y(t)$ for all $t \geq t_0$.

A Taxonomy of Systems

State?

- The *state* $x(t_0)$ of a system at the time instant t_0 is the information that together with the input $u(t)$ for $t \geq t_0$ univocally determines the output $y(t)$ for all $t \geq t_0$.
- The state $x(t_0)$ summarises all the system history from $t = -\infty$ to t_0, e.g. with the knowledge of the angle θ and the angular velocity $\dot{\theta}$ at time t_0, we can predict the response of the robot arm to torque inputs u for all time $t \geq t_0$.

A Taxonomy of Systems

State?

- The *state* $x(t_0)$ of a system at the time instant t_0 is the information that together with the input $u(t)$ for $t \geq t_0$ univocally determines the output $y(t)$ for all $t \geq t_0$.
- The state $x(t_0)$ summarises all the system history from $t = -\infty$ to t_0, e.g. with the knowledge of the angle θ and the angular velocity $\dot{\theta}$ at time t_0, we can predict the response of the robot arm to torque inputs u for all time $t \geq t_0$.
- The *input* at $t \geq t_0$ and the *initial conditions* $x(t_0)$ determine the evolution of the system for $t \geq t_0$, which we could represent as

$$y(t),\ t \geq t_0 \Lleftarrow \begin{cases} x(t_0) \\ u(t),\ t \geq t_0 \end{cases}$$

A Taxonomy of Systems

Finite-dimensional?

- Means that the state $x(t)$ at any given instant of time t can be completely characterised by a finite number of parameters.

A Taxonomy of Systems

Finite-dimensional?

- Means that the state $x(t)$ at any given instant of time t can be completely characterised by a finite number of parameters.

- In the case of the robot arm, two parameters: angle θ and angular velocity $\dot{\theta}$.

A Taxonomy of Systems

Continuous-time?

- Means that the independent variable, time t, takes values in a *continuum*, the set of real numbers $\mathbb{R}$.

A Taxonomy of Systems

Continuous-time?

- Means that the independent variable, time t, takes values in a *continuum*, the set of real numbers $\mathbb{R}$.

- In contrast, a system defined by a *difference equation*, like

$$x[k+1] = Ax[k] + Bu[x],$$

the independent variable k can, for example, take values only in the set of integers $\mathbb{N}$, $k = \cdots - 1, 0, 1, 2 \ldots.$

Linear Systems

- A system is said to be linear if it satisfies the *superposition principle*, that is, if given two pairs of initial conditions and inputs,

$$y_i(t),\ t \geq t_0 \Leftarrow \begin{cases} x_i(t_0) \\ u_i(t),\ t \geq t_0 \end{cases} \qquad \text{for } i = 1, 2\,,$$

Linear Systems

- A system is said to be linear if it satisfies the *superposition principle*, that is, if given two pairs of initial conditions and inputs,

$$y_i(t),\ t \geq t_0 \Lleftarrow \begin{cases} x_i(t_0) \\ u_i(t),\ t \geq t_0 \end{cases} \qquad \text{for } i = 1, 2\,,$$

then we have that

$$y_1(t) + y_2(t),\ t \geq t_0 \Lleftarrow \begin{cases} x_1(t_0) + x_2(t_0) \\ u_1(t) + u_2(t),\ t \geq t_0 \end{cases} \qquad \text{(additivity)}$$

$$\alpha y_i(t),\ t \geq t_0 \Lleftarrow \begin{cases} \alpha x_i(t_0) \\ \alpha u_i(t),\ t \geq t_0 \end{cases} \quad \alpha \in \mathbb{R} \qquad \text{(homogeneity)}$$

Linear Systems

- The combination of the properties of *additivity* and that of *homogeneity* yields the property of *superposition*.

Linear Systems

- The combination of the properties of *additivity* and that of *homogeneity* yields the property of *superposition*.
- A system that does not satisfy the property of superposition is nonlinear.

Linear Systems

- The combination of the properties of *additivity* and that of *homogeneity* yields the property of *superposition*.
- A system that does not satisfy the property of superposition is nonlinear.
- By the property of additivity we can consider the response of the system to initial conditions independently from that due to inputs.

$$y(t) = y_l(t) + y_f(t),\ t \geq t_0 \Leftarrow \begin{cases} y_l(t),\ t \geq t_0 & \Leftarrow \begin{cases} x(t_0) \\ u(t) = 0,\ t \geq t_0 \end{cases} \\ y_f(t),\ t \geq t_0 & \Leftarrow \begin{cases} x(t_0) = 0 \\ u(t),\ t \geq t_0 \end{cases} \end{cases}$$

Linear Systems

The response of a linear system is the superposition of its *free* response (that to initial conditions only, without external input) and its *forced* response (that to an external input, with zero initial conditions).

Linear Time-Invariant Systems

- A system is *time-invariant* if for each pair of initial conditions and inputs

$$y(t),\ t \geq t_0 \Lleftarrow \begin{cases} x(t_0) \\ u(t),\ t \geq t_0 \end{cases}$$

and each $T \in \mathbb{R}$, we have that

$$y(t-T),\ t \geq t_0 + T \Lleftarrow \begin{cases} x(t_0 + T) \\ u(t-T),\ t \geq t_0 + T\,. \end{cases}$$

Linear Time-Invariant Systems

- A system is *time-invariant* if for each pair of initial conditions and inputs

$$y(t),\ t \geq t_0 \Lleftarrow \begin{cases} x(t_0) \\ u(t),\ t \geq t_0 \end{cases}$$

and each $T \in \mathbb{R}$, we have that

$$y(t-T),\ t \geq t_0 + T \Lleftarrow \begin{cases} x(t_0 + T) \\ u(t-T),\ t \geq t_0 + T\,. \end{cases}$$

- In other words, the system gives the same response, but shifted in time, that if we apply to it the same input shifted in time, while keeping the same initial conditions.

Linear Time-Invariant Systems

- A system is *time-invariant* if for each pair of initial conditions and inputs

$$y(t),\ t \geq t_0 \Leftarrow \begin{cases} x(t_0) \\ u(t),\ t \geq t_0 \end{cases}$$

and each $T \in \mathbb{R}$, we have that

$$y(t-T),\ t \geq t_0 + T \Leftarrow \begin{cases} x(t_0 + T) \\ u(t-T),\ t \geq t_0 + T\,. \end{cases}$$

- In other words, the system gives the same response, but shifted in time, that if we apply to it the same input shifted in time, while keeping the same initial conditions.
- A system without this property is called *time-varying*.

Linear Time-Invariant Systems

Input-Output Representation

- From the superposition principle, we can obtain the representation of a linear system by the *convolution integral*

$$y(t) = \int_{-\infty}^{\infty} g(t, \tau) u(\tau)\, d\tau\,, \tag{1}$$

where $g(t, \tau)$ is the *impulse response* of the system, that is, the output produced by a unitary impulse $\delta(t)$ applied at the input at the time instant τ.

Linear Time-Invariant Systems

Input-Output Representation

- From the superposition principle, we can obtain the representation of a linear system by the *convolution integral*

$$y(t) = \int_{-\infty}^{\infty} g(t,\tau)u(\tau)\, d\tau\,, \qquad (2)$$

  where $g(t,\tau)$ is the *impulse response* of the system, that is, the output produced by a unitary impulse $\delta(t)$ applied at the input at the time instant τ.

- Causality implies that

$$\text{causality} \Leftrightarrow g(t,\tau) = 0 \text{ for } t < \tau,$$

  and on assuming zero initial conditions, Equation (1) then yields

$$y(t) = \int_{t_0}^{t} g(t,\tau)u(\tau)\, d\tau\,.$$

Linear Time-Invariant Systems

Input-Output Representation

- When the system has p inputs and q outputs, then we use the *impulse response matrix* $G(t,\tau) \in \mathbb{R}^{q \times p}$.

Linear Time-Invariant Systems

Input-Output Representation

- When the system has p inputs and q outputs, then we use the *impulse response matrix* $G(t,\tau) \in \mathbb{R}^{q \times p}$.
- If the system is time-invariant, then for any T we have that

$$g(t,\tau) = g(t+T, \tau+T) = g(t-\tau, 0),$$

and we can redefine $g(t-\tau, 0)$ simply as $g(t-\tau)$. Thus the input-output representation of the system reduces to

$$y(t) = \int_0^t g(t-\tau)u(\tau)\, d\tau = \int_0^t g(\tau)u(t-\tau)\, d\tau\,.$$

Linear Time-Invariant Systems

Input-Output Representation

- When the system has p inputs and q outputs, then we use the *impulse response matrix* $G(t,\tau) \in \mathbb{R}^{q \times p}$.
- If the system is time-invariant, then for any T we have that

$$g(t,\tau) = g(t+T,\tau+T) = g(t-\tau,0)\,,$$

  and we can redefine $g(t-\tau,0)$ simply as $g(t-\tau)$. Thus the input-output representation of the system reduces to

$$y(t) = \int_0^t g(t-\tau)u(\tau)\,d\tau = \int_0^t g(\tau)u(t-\tau)\,d\tau\,.$$

- The condition of causality for a linear time-invariant system can be alternatively stated as $g(t) = 0$ for $t < 0$.

Linear Time-Invariant Systems

State Space Representation

- Every linear finite-dimensional system can be described by *state space equations*

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t). \end{aligned} \tag{3}$$

Linear Time-Invariant Systems

State Space Representation

- Every linear finite-dimensional system can be described by *state space equations*

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \,. \end{aligned} \tag{4}$$

- For a system with *order* n, the *state vector* is a vector of dimensions $n \times 1$, that is, it stacks n state variables, $x(t) \in \mathbb{R}^n$, for every t. If the system has p inputs and q outputs, then $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^q$.

Linear Time-Invariant Systems

State Space Representation

- Every linear finite-dimensional system can be described by *state space equations*

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t)\\ y(t) &= C(t)x(t) + D(t)u(t)\,.\end{aligned} \tag{5}$$

- For a system with *order* n, the *state vector* is a vector of dimensions $n \times 1$, that is, it stacks n state variables, $x(t) \in \mathbb{R}^n$, for every t. If the system has p inputs and q outputs, then $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^q$.
- The matrices A, B, C, D are usually called

$$\begin{aligned}A &\in \mathbb{R}^{n\times n} : \textit{evolution matrix}\\ B &\in \mathbb{R}^{n\times p} : \textit{input matrix}\\ C &\in \mathbb{R}^{q\times n} : \textit{output matrix}\\ D &\in \mathbb{R}^{q\times p} : \textit{direct feedthrough matrix}\end{aligned}$$

Linear Time-Invariant Systems

State Space Representation

- When, in addition, the system is time-invariant, then the state space representation (3) reduces to

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\,. \end{aligned} \tag{6}$$

Linear Time-Invariant Systems

State Space Representation

- When, in addition, the system is time-invariant, then the state space representation (3) reduces to

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\,. \end{aligned} \tag{8}$$

- By applying the Laplace transform to (6) we obtain

$$\begin{aligned} s\hat{x}(s) - x(0) &= A\hat{x}(s) + B\hat{u}(s) \\ \hat{y}(s) &= C\hat{x}(s) + D\hat{u}(s)\,, \end{aligned}$$

from which follow

$$\begin{aligned} \hat{x}(s) &= (sI - A)^{-1}x(0) + (sI - A)^{-1}B\hat{u}(s) \\ \hat{y}(s) &= C(sI - A)^{-1}x(0) + [C(sI - A)^{-1}B + D]\hat{u}(s)\,. \end{aligned} \tag{9}$$

Linear Time-Invariant Systems

State Space Representation

- The algebraic equations (7) allow us to compute $\hat{x}(s)$ and $\hat{y}(s)$ from $x(0)$ and $\hat{u}(s)$. Then the inverse Laplace transform will give $x(t)$ and $y(t)$. By letting $x(0) = 0$ we see that the transfer function of the system is

$$\hat{G}(s) = C(sI - A)^{-1}B + D$$

Linear Time-Invariant Systems

State Space Representation

- The algebraic equations (7) allow us to compute $\hat{x}(s)$ and $\hat{y}(s)$ from $x(0)$ and $\hat{u}(s)$. Then the inverse Laplace transform will give $x(t)$ and $y(t)$. By letting $x(0) = 0$ we see that the transfer function of the system is
$$\hat{G}(s) = C(sI - A)^{-1}B + D$$

- In MATLAB the functions `tf2ss` and `ss2tf` allow us to convert from and to one representation to the other.

Linear Time-Invariant Systems

State Space Representation

- The algebraic equations (7) allow us to compute $\hat{x}(s)$ and $\hat{y}(s)$ from $x(0)$ and $\hat{u}(s)$. Then the inverse Laplace transform will give $x(t)$ and $y(t)$. By letting $x(0) = 0$ we see that the transfer function of the system is

$$\hat{G}(s) = C(sI - A)^{-1}B + D$$

- In MATLAB the functions `tf2ss` and `ss2tf` allow us to convert from and to one representation to the other.
- See also the functions `ss`,`tf`,`ssdata` and `tfdata`, for system representations in MATLAB.

Linearisation

- Most physical systems are nonlinear. An important class of them can be represented by state space equations in the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), x(t_0), t)\,, \quad x(t_0) = x_0 \\ y(t) &= h(x(t), u(t), x(t_0), t)\,, \end{aligned} \qquad (10)$$

where f and h are nonlinear vector fields, that is, in scalar terms, the i-component of $\dot{x}(t)$ in (10) is written as

$$\dot{x}_i(t) = f_i(x_1(t), \dots, x_n(t); u_1(t), \dots, u_m(t); x_1(t_0), \dots, x_n(t_0); t) \quad x_i(t_0) = x_{i0}\,.$$

Linearisation

- Most physical systems are nonlinear. An important class of them can be represented by state space equations in the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), x(t_0), t)\,, \quad x(t_0) = x_0 \\ y(t) &= h(x(t), u(t), x(t_0), t)\,, \end{aligned} \tag{11}$$

where f and h are nonlinear vector fields, that is, in scalar terms, the i-component of $\dot{x}(t)$ in (10) is written as

$$\dot{x}_i(t) = f_i(x_1(t), \ldots, x_n(t); u_1(t), \ldots, u_m(t); x_1(t_0), \ldots, x_n(t_0); t) \quad x_i(t_0) = x_{i0}\,.$$

- A linear state space equation is a useful tool to describe systems like (10) in an *approximate* way.

Linearisation

- Most physical systems are nonlinear. An important class of them can be represented by state space equations in the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), x(t_0), t)\,, \quad x(t_0) = x_0 \\ y(t) &= h(x(t), u(t), x(t_0), t)\,, \end{aligned} \tag{12}$$

  where f and h are nonlinear vector fields, that is, in scalar terms, the i-component of $\dot{x}(t)$ in (10) is written as

$$\dot{x}_i(t) = f_i(x_1(t), \dots, x_n(t); u_1(t), \dots, u_m(t); x_1(t_0), \dots, x_n(t_0); t) \quad x_i(t_0) = x_{i0}\,.$$

- A linear state space equation is a useful tool to describe systems like (10) in an *approximate* way.
- The process of obtaining a linear model from a nonlinear one is called *linearisation*.

Linearisation

- The linearisation is performed around a nominal *point* or *trajectory*, defined by *nominal* values $\tilde{x}(t)$, $\tilde{x}_0$ and $\tilde{u}(t)$ that satisfy (10),

$$\tilde{x}(t),\ t \geq t_0 \Lleftarrow \begin{cases} \tilde{x}(t_0) \\ \tilde{u}(t),\ t \geq t_0 \end{cases}$$

Linearisation

- The linearisation is performed around a nominal *point* or *trajectory*, defined by *nominal* values $\tilde{x}(t)$, $\tilde{x}_0$ and $\tilde{u}(t)$ that satisfy (10),

$$\tilde{x}(t),\ t \geq t_0 \Lleftarrow \begin{cases} \tilde{x}(t_0) \\ \tilde{u}(t),\ t \geq t_0 \end{cases}$$

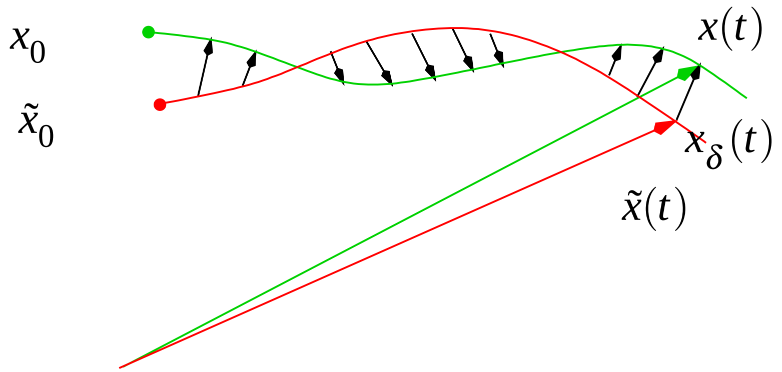
- We are interested in the behaviour of the nonlinear differential equation (10) for an input and initial state which are “close” to the nominal values, that is, $u(t) = \tilde{u}(t) + u_\delta(t)$ and $x_0 = \tilde{x}_0 + x_{0\delta}$ for $u_\delta(t)$ and $x_{0\delta}$ sufficiently small for all $t \geq t_0$.

Linearisation

- The linearisation is performed around a nominal *point* or *trajectory*, defined by *nominal* values $\tilde{x}(t)$, $\tilde{x}_0$ and $\tilde{u}(t)$ that satisfy (10),

$$\tilde{x}(t),\ t \geq t_0 \Lleftarrow \begin{cases} \tilde{x}(t_0) \\ \tilde{u}(t),\ t \geq t_0 \end{cases}$$

- We are interested in the behaviour of the nonlinear differential equation (10) for an input and initial state which are "close" to the nominal values, that is, $u(t) = \tilde{u}(t) + u_\delta(t)$ and $x_0 = \tilde{x}_0 + x_{0\delta}$ for $u_\delta(t)$ and $x_{0\delta}$ sufficiently small for all $t \geq t_0$.



Linearisation

- Suppose that the solution stays close to the nominal trajectory, and write $x(t) = \tilde{x}(t) + x_\delta(t)$ for each $t \geq t_0$.

Linearisation

- Suppose that the solution stays close to the nominal trajectory, and write $x(t) = \tilde{x}(t) + x_\delta(t)$ for each $t \geq t_0$.
- In terms of the nonlinear state space equation (10) we have

$$\dot{\tilde{x}}(t) + \dot{\tilde{x}}_\delta(t) = f(\tilde{x}(t) + x_\delta(t), \tilde{u}(t) + u_\delta(t), t), \quad \tilde{x}(t_0) + x_\delta(t_0) = \tilde{x}_0 + x_{0\delta} \quad (15)$$

Linearisation

- Suppose that the solution stays close to the nominal trajectory, and write $x(t) = \tilde{x}(t) + x_\delta(t)$ for each $t \geq t_0$.
- In terms of the nonlinear state space equation (10) we have

$$\dot{\tilde{x}}(t)+\dot{\tilde{x}}_\delta(t) = f(\tilde{x}(t)+x_\delta(t), \tilde{u}(t)+u_\delta(t), t), \quad \tilde{x}(t_0)+x_\delta(t_0) = \tilde{x}_0+x_{0\delta} \quad (17)$$

- Assuming differentiability, we can expand the right hand side of (13) in Taylor series around $\tilde{x}(t)$ and $\tilde{u}(t)$, keeping only the first order terms. *Note* that the expansion is performed in terms of x and u, and not for the independent variable t.

Linearisation

- Suppose that the solution stays close to the nominal trajectory, and write $x(t) = \tilde{x}(t) + x_\delta(t)$ for each $t \geq t_0$.
- In terms of the nonlinear state space equation (10) we have

$$\dot{\tilde{x}}(t)+\dot{\tilde{x}}_\delta(t) = f(\tilde{x}(t)+x_\delta(t), \tilde{u}(t)+u_\delta(t), t), \quad \tilde{x}(t_0)+x_\delta(t_0) = \tilde{x}_0+x_{0\delta} \quad (19)$$

- Assuming differentiability, we can expand the right hand side of (13) in Taylor series around $\tilde{x}(t)$ and $\tilde{u}(t)$, keeping only the first order terms. *Note* that the expansion is performed in terms of x and u, and not for the independent variable t.
- We make the operation more explicit for the i-component, which yields

$$f_i(\tilde{x} + x_\delta, \tilde{u} + u_\delta, t) \approx f_i(\tilde{x}, \tilde{u}, t) + \frac{\partial f_i}{\partial x_1}(\tilde{x}, \tilde{u}, t)x_{\delta 1} + \cdots + \frac{\partial f_i}{\partial x_n}(\tilde{x}, \tilde{u}, t)x_{\delta n}$$
$$+ \frac{\partial f_i}{\partial u_1}(\tilde{x}, \tilde{u}, t)u_{\delta 1} + \cdots + \frac{\partial f_i}{\partial u_m}(\tilde{x}, \tilde{u}, t)u_{\delta m} \quad (20)$$

Linearisation

- By repeating this operation for each $i = 1, \ldots, n$, and returning to the vectorial notation, we have

$$\dot{\tilde{x}}(t) + \dot{\tilde{x}}_\delta(t) \approx f(\tilde{x}(t), \tilde{u}(t)) + \frac{\partial f}{\partial x}(\tilde{x}, \tilde{u}, t)x_\delta + \frac{\partial f}{\partial u}(\tilde{x}, \tilde{u}, t)u_\delta$$

where $\frac{\partial f}{\partial x}$ represents the *Jacobian*, or *Jacobian Matrix*, of the vector field f with respect to x,

$$\frac{\partial f}{\partial x} \triangleq \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \ldots & \ldots & \ldots & \ldots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \ldots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Linearisation

- Since $\dot{\tilde{x}}(t) = f(\tilde{x}(t), \tilde{u}(t), t)$, $\tilde{x}(t_0) = \tilde{x}_0$, the relation between $x_\delta(t)$ and $u_\delta(t)$ (the *incremental model*) is approximately described by a linear, *time-varying* state equation of the form

$$\dot{x}_\delta(t) = A(t)x_\delta(t) + B(t)u_\delta(t), \quad x_\delta(t_0) = x_0 - \tilde{x}_0$$

where

$$A(t) = \frac{\partial f}{\partial x}(\tilde{x}(t), \tilde{u}(t), t), \quad B(t) = \frac{\partial f}{\partial u}(\tilde{x}(t), \tilde{u}(t), t).$$

Linearisation

- Since $\dot{\tilde{x}}(t) = f(\tilde{x}(t), \tilde{u}(t), t)\,, \quad \tilde{x}(t_0) = \tilde{x}_0$, the relation between $x_\delta(t)$ and $u_\delta(t)$ (the *incremental model*) is approximately described by a linear, *time-varying* state equation of the form

$$\dot{x}_\delta(t) = A(t)x_\delta(t) + B(t)u_\delta(t), \quad x_\delta(t_0) = x_0 - \tilde{x}_0$$

  where

$$A(t) = \frac{\partial f}{\partial x}(\tilde{x}(t), \tilde{u}(t), t)\,, \quad B(t) = \frac{\partial f}{\partial u}(\tilde{x}(t), \tilde{u}(t), t)\,.$$

- In the same way we can expand the output equation $y(t) = h(x(t), u(t), t)$, from which we obtain the linear approximation

$$y_\delta(t) = C(t)x_\delta(t) + D(t)u_\delta(t),$$

  where $y_\delta(t) = y(t) - \tilde{y}(t)$, with $\tilde{y}(t) = h(\tilde{x}(t), \tilde{u}(t), t)$ and

$$C(t) = \frac{\partial h}{\partial x}(\tilde{x}(t), \tilde{u}(t), t)\,, \quad D(t) = \frac{\partial h}{\partial u}(\tilde{x}(t), \tilde{u}(t), t)\,.$$

Linearisation

Note that the state equations obtained by linearisation will in general be *time-varying*, even when the original vector fields f and h were time-invariant, because the Jacobian matrices are evaluated along trajectories, and not stationary points.

Discrete-Time Systems

- Most of the state space concepts for linear continuous-time systems can be directly translated to discrete-time systems, described by *linear difference equations*. In this case the time variable t only takes values on a denumerable set, like the integers.

Discrete-Time Systems

- Most of the state space concepts for linear continuous-time systems can be directly translated to discrete-time systems, described by *linear difference equations*. In this case the time variable t only takes values on a denumerable set, like the integers.
- When the discrete-time system is obtained from sampling a continuous-time system, we will only consider *regular* sampling, where $t = kT$, $k = 0, 1, 2, \dots$, and T is the *sampling period*. In this case we denote the discrete-time variables (sequences) as $u[k] \triangleq u(kT)$, etc.

Discrete-Time Systems

- Most of the state space concepts for linear continuous-time systems can be directly translated to discrete-time systems, described by *linear difference equations*. In this case the time variable t only takes values on a denumerable set, like the integers.
- When the discrete-time system is obtained from sampling a continuous-time system, we will only consider *regular* sampling, where $t = kT$, $k = 0, 1, 2, \dots$, and T is the *sampling period*. In this case we denote the discrete-time variables (sequences) as $u[k] \triangleq u(kT)$, etc.
- The concepts of finite dimensionality, causality, linearity and the superposition principle for responses to initial conditions and inputs are exactly the same as those in the continuous-time case.

Discrete-Time Systems

- Most of the state space concepts for linear continuous-time systems can be directly translated to discrete-time systems, described by *linear difference equations*. In this case the time variable t only takes values on a denumerable set, like the integers.
- When the discrete-time system is obtained from sampling a continuous-time system, we will only consider *regular* sampling, where $t = kT$, $k = 0, 1, 2, \dots$, and T is the *sampling period*. In this case we denote the discrete-time variables (sequences) as $u[k] \triangleq u(kT)$, etc.
- The concepts of finite dimensionality, causality, linearity and the superposition principle for responses to initial conditions and inputs are exactly the same as those in the continuous-time case.
- One difference though: pure delays in discrete-time do not give rise to an infinite-dimensional system, as is the case of continuous-time systems, if the delay is a multiple of the sampling period T.

Discrete-Time Systems

Input-Output Representation

- We define the *impulse sequence* $\delta[k]$ as

$$\delta[k-m] = \begin{cases} 1 & \text{if } k = m \\ 0 & \text{if } k \neq m \end{cases}$$

where k and m are integers.

Discrete-Time Systems

Input-Output Representation

- We define the *impulse sequence* $\delta[k]$ as

$$\delta[k-m] = \begin{cases} 1 & \text{if } k = m \\ 0 & \text{if } k \neq m \end{cases}$$

  where k and m are integers.

- Note how in the discrete-time case impulses are easy to implement physically, in contrast to the continuous-time case.

Discrete-Time Systems

Input-Output Representation

- We define the *impulse sequence* $\delta[k]$ as

$$\delta[k-m] = \begin{cases} 1 & \text{if } k = m \\ 0 & \text{if } k \neq m \end{cases}$$

  where k and m are integers.

- Note how in the discrete-time case impulses are easy to implement physically, in contrast to the continuous-time case.

- In a discrete-time linear system every input sequence $u[k]$ can be represented by means of the series

$$u[k] = \sum_{m=-\infty}^{\infty} u[m]\delta[k-m].$$

Discrete-Time Systems

Input-Output Representation

- If $g[k, m]$ denotes the output of a discrete time system to an impulse sequence applied at the instant m, then we have that

$$\delta[k - m] \rightarrow g[k, m]$$

$$\delta[k, m]u[m] \rightarrow g[k, m]u[m] \qquad \text{(by homogeneity)}$$

$$\sum_m \delta[k, m]u[m] \rightarrow \sum_m g[k, m]u[m] \qquad \text{(by additivity)}.$$

Discrete-Time Systems

Input-Output Representation

- If $g[k, m]$ denotes the output of a discrete time system to an impulse sequence applied at the instant m, then we have that

$$\begin{aligned} \delta[k-m] &\rightarrow g[k,m] \\ \delta[k,m]u[m] &\rightarrow g[k,m]u[m] \qquad \text{(by homogeneity)} \\ \sum_m \delta[k,m]u[m] &\rightarrow \sum_m g[k,m]u[m] \qquad \text{(by additivity)}. \end{aligned}$$

- Thus the output $y[k]$ obtained from the input $u[k]$ can be written by means of the series

$$y[k] = \sum_{m=-\infty}^{\infty} g[k,m]u[m]. \tag{22}$$

Discrete-Time Systems

Input-Output Representation

- If the system is *causal* there wouldn't be output signal before the input is applied, hence

$$\text{causality} \Leftrightarrow g[k, m] = 0 \text{ for } k < m.$$

Discrete-Time Systems

Input-Output Representation

- If the system is *causal* there wouldn't be output signal before the input is applied, hence

$$\text{causality} \Leftrightarrow g[k, m] = 0 \text{ for } k < m.$$

- For *causal* discrete-time systems the representation (21) reduces to

$$y[k] = \sum_{m=k_0}^{k} g[k, m]u[m]\,,$$

and, if in addition we have *time-invariance*, the property of invariance with respect to shifts in time holds, and thus we arrive to the system representation by the *discrete convolution*

$$y[k] = \sum_{m=0}^{k} g[k - m]u[m] = \sum_{m=0}^{k} g[m]u[k - m]\,.$$

Discrete-Time Systems

State Space Representation

- Every discrete-time, finite dimensional, linear system can be represented by state space difference equations, as in

$$x[k+1] = A[k]x[k] + B[k]u[k]$$
$$y[k] = C[k]x[k] + D[k]u[k],$$

and in the time-invariant case

$$x[k+1] = Ax[k] + Bu[k]$$
$$y[k] = Cx[k] + Du[k].$$

Discrete-Time Systems

State Space Representation

- In this case, it corresponds to talk about *discrete* transfer functions, $\hat{G}(z) = \mathcal{Z}[g[k]]$. The relation between discrete transfer function representation and state space representation is identical to the continuous-time case,

$$\hat{G}(z) = C(zI - A)^{-1}B + D\,,$$

and the same MATLAB functions can be used.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.
- Infinite-dimensional systems cannot be represented by state space equations (they would require an infinite number of them).

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.
- Infinite-dimensional systems cannot be represented by state space equations (they would require an infinite number of them).
- By linearisation, one can describe the behaviour of a nonlinear system *approximately* by means of an incremental linear state space model.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.
- Infinite-dimensional systems cannot be represented by state space equations (they would require an infinite number of them).
- By linearisation, one can describe the behaviour of a nonlinear system *approximately* by means of an incremental linear state space model.
- Linearisation is performed around a known *nominal trajectory*, and the incremental models obtained will, in general, be *time-varying*.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.
- Infinite-dimensional systems cannot be represented by state space equations (they would require an infinite number of them).
- By linearisation, one can describe the behaviour of a nonlinear system *approximately* by means of an incremental linear state space model.
- Linearisation is performed around a known *nominal trajectory*, and the incremental models obtained will, in general, be *time-varying*.
- Discrete-time systems have representations equivalent to those of continuous-time systems by *convolution series*, transfer functions in the discrete $\mathcal{Z}$ transform, and state space difference equations.

A Few General Facts to Remember

- A transfer matrix is *rational* if and only if the corresponding system is linear, time-invariant and finite-dimensional.
- Input-output representations usually assume zero initial conditions.
- Infinite-dimensional systems cannot be represented by state space equations (they would require an infinite number of them).
- By linearisation, one can describe the behaviour of a nonlinear system *approximately* by means of an incremental linear state space model.
- Linearisation is performed around a known *nominal trajectory*, and the incremental models obtained will, in general, be *time-varying*.
- Discrete-time systems have representations equivalent to those of continuous-time systems by *convolution series*, transfer functions in the discrete $\mathcal{Z}$ transform, and state space difference equations.
- In contrast to the continuous time case, pure delays do not necessarily give rise to an infinite-dimensional discrete-time system.

A Few General Facts to Remember

Type of system	Internal representation	External representation
infinite dim. linear		$y(t) = \int_{t_0}^{t} G(t,\tau)u(\tau)d\tau$
finite dim., linear	$\dot{x} = A(t)x + B(t)u$ $y = C(t)x + D(t)u$	$y(t) = \int_{t_0}^{t} G(t,\tau)u(\tau)d\tau$
infinite dim. LTI		$y(t) = \int_{t_0}^{t} G(t,\tau)u(\tau)d\tau$ $\hat{y}(s) = \hat{G}(s)\hat{u}(s)$
finite dim., LTI	$\dot{x} = Ax + Bu$ $y = Cx + Du$	$y(t) = \int_{t_0}^{t} G(t,\tau)u(\tau)d\tau$ $\hat{y}(s) = \hat{G}(s)\hat{u}(s)$