A Primer on Dynamic Systems

(Kalman Filtering Lecture Prerequisite)

Reza Sameni

Department of Biomedical Informatics
Emory University, Atlanta, GA, USA
Email: rsameni@dbmi.emory.edu

July 2021

Background

Signal

Any waveform or time-series represented as a function of an independent variable (time, space, samples, etc.).

Example

Blood pressure variations in time, change of pressure vs altitude, stock market daily charts, electrocardiogram and electroencephalogram waveforms, etc.

System

- **General definition:** a set of entities interacting together as parts of a mechanism forming a complex whole.
- **In our context:** a mathematical/algorithmic process that maps an input to an output.

Example

Cardiovascular system, thermal regulation system, glucose control system, etc.

System Properties: Structure

- **Static:** No temporal evolution in the system or its output

Example

$$y = x^2 + 2x + \sqrt{x} \tag{1}$$

- **Dynamic:** The system evolves in time, i.e., the input-output process is not point-wise; besides the instantaneous input, the output also depends on the rate changes in the input

Example

$$\ddot{y}(t) + 2\alpha\dot{y}(t) + \omega_0^2 y(t) = x(t) \tag{2}$$

System Properties
(continued)

Linearity

- **Linear:** The input-output relationship is linear: additivity and homogeneity hold
- **Nonlinear:** The input-output relationship is nonlinear (the system is either non-additive or non-homogeneous)

Time Dependence

- **Time-variant:** The input-output mapping depends on the time origin
- **Time-invariant:** The input-output mapping *does not* depend on the time origin

Input-Output Mathematical Description

Dynamic systems are of great interest in biological systems modeling. The input-output relationship of these systems can be expressed as follows:

- **Explicit:** The output can be directly calculated from the input

Example

$$y(t) = x^2(t) + kt \tag{3}$$

- **Implicit:** Only an implicit relation between the input and output are available

Example

$$\ddot{y}(t) + \alpha\dot{y}(t) + b\sqrt{y(t)} = x(t) \tag{4}$$

Differential Equations

In many systems, the *rate* of output and input of a system are related (instead of the input or outputs themselves). This relationship can be modeled by differential equations. The general form of a linear differential equation is:

$$\sum_{k=0}^{N-1} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M-1} b_k \frac{d^k x(t)}{dt^k} \tag{5}$$

- If the coefficients in (5) are constant, the system is time-invariant
- If the initial states are zero, the system is linear; otherwise it is incrementally linear (affine)

Differential Equations
(continued)

The solution of (5) is

$$y(t) = y_h(t) + y_p(t) \tag{6}$$

where

- $y_h(t)$: the homogeneous response to initial states
- $y_p(t)$: the particular response to the system input

The system is only linear when $y_h(t) = 0$, i.e., initial states are zero

Transfer Functions

For linear time-invariant systems, using Laplace (or Fourier) transform, a transfer function can be assigned to the system:

$$H(s) \triangleq \frac{Y(s)}{X(s)} \tag{7}$$

where $X(s)$ and $Y(s)$ are Laplace transforms of $x(t)$ and $y(t)$.

- $H(s)$ is used for *steady-state* analysis but not (explicitly) for *transient analysis*. For *stable* systems, the response to initial states converge to zero. Hence, in steady-state, the output can be exactly determined using $H(s)$
- The system is stable if the poles are in the left half plane.
- The farther the poles are from the origin, the faster the effect of initial states vanish

Transfer Functions
(continued)

Transfer Function Limitations

- It is restricted to LTI systems (or systems which can be approximated by LTI systems)
- It can only be used for steady-state analysis and and not for transient analysis (although the transfer function has indirect implications on the transient properties of a system)

Transfer Functions
(continued)

Transfer Function Limitations

- It is restricted to LTI systems (or systems which can be approximated by LTI systems)
- It can only be used for steady-state analysis and and not for transient analysis (although the transfer function has indirect implications on the transient properties of a system)

Alternative Approach: *State-space analysis*

State Space

States

- States are properties of a system (usually represented as a vector), through which the system's response to a given input can be uniquely found. States are properties of a system, knowing which make us needless of its past.
- In a physical system, storage of the 'past information' requires energy saving elements (capacitors, inductors, neurons, cells, etc.). In digital systems memory elements (registers) are required to preserve states
- State variables are not necessarily unique; but for linear systems their total number (known as the *system's order*) is fixed
- A system's order can be infinite

State Variables
(continued)

Example

$$\dot{y}(t) + \alpha y(t) = \beta x(t) \tag{8}$$

State Variables
(continued)

Example

$$\dot{y}(t) + \alpha y(t) = \beta x(t) \quad (8)$$

This system has one state; knowledge of $y(t_0)$ is sufficient to determine $y(t)$ for all $t \in [t_0, \infty)$.

State Variables
(continued)

Example

$$\dot{y}(t) + \alpha y(t) = \beta x(t) \tag{8}$$

This system has one state; knowledge of $y(t_0)$ is sufficient to determine $y(t)$ for all $t \in [t_0, \infty)$.

Example

$$\begin{aligned} &y(t) = x(t - t_0) \quad \text{or} \quad H(s) = e^{-t_0 s} \\ &H(s) = 1 - st_0 + \frac{(st_0)^2}{2} - \cdots = \frac{1}{1 + st_0 + \frac{(st_0)^2}{2} + \cdots} \end{aligned} \tag{9}$$

State Variables
(continued)

Example

$$\dot{y}(t) + \alpha y(t) = \beta x(t) \tag{8}$$

This system has one state; knowledge of $y(t_0)$ is sufficient to determine $y(t)$ for all $t \in [t_0, \infty)$.

Example

$$\begin{aligned} & y(t) = x(t - t_0) \quad \text{or} \quad H(s) = e^{-t_0 s} \\ & H(s) = 1 - st_0 + \frac{(st_0)^2}{2} - \cdots = \frac{1}{1 + st_0 + \frac{(st_0)^2}{2} + \cdots} \end{aligned} \tag{9}$$

This system has infinite number of states. For $|st_0| \ll 1$, the system can be approximated with lower order systems. A small value of st_0 implies a *narrow-band* system working at a relatively *low frequency*.

State Space Representations

State-space equations

Set of first-order differential (difference) equations relating the system inputs, outputs, and states

$$\begin{cases} \dot{\boldsymbol{s}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{s}(t)) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{s}(t)) \end{cases} \tag{10}$$

where $\boldsymbol{x}(t)$, $\boldsymbol{y}(t)$, and $\boldsymbol{s}(t)$ are the input, output and state vectors.

State Space Representations

State-space equations

Set of first-order differential (difference) equations relating the system inputs, outputs, and states

$$\begin{cases} \dot{\boldsymbol{s}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{s}(t)) \\ \boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{s}(t)) \end{cases} \tag{10}$$

where $\boldsymbol{x}(t)$, $\boldsymbol{y}(t)$, and $\boldsymbol{s}(t)$ are the input, output and state vectors.

- State-space representations are not unique; but some *canonical forms* are more common
- For LTI systems, differential and state-space equations can be converted to one another
- Scalar differential equations of order n, are converted into n first order state-space differential equations (if the system is irreducible)

State Space Representations
(continued)

State-space equations for multi-input multi-output (MIMO) systems

$$\begin{cases} \dot{s}_1 = f_1(s_1, \cdots, s_n, x_1, \cdots, x_p) \\ \vdots \\ \dot{s}_n = f_n(s_1, \cdots, s_n, x_1, \cdots, x_p) \end{cases} \tag{11}$$

$$\begin{cases} y_1 = g_1(s_1, \cdots, s_n, x_1, \cdots, x_p) \\ \vdots \\ y_m = g_m(s_1, \cdots, s_n, x_1, \cdots, x_p) \end{cases} \tag{12}$$

State Space Representations
(continued)

Linear Case

$$\mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix}, \mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \tag{13}$$

$$\begin{cases} \dot{\mathbf{s}}_{n\times 1} = \mathbf{A}_{n\times n}\mathbf{s}_{n\times 1} + \mathbf{B}_{n\times p}\mathbf{x}_{p\times 1} \\ \mathbf{y}_{m\times 1} = \mathbf{C}_{m\times n}\mathbf{s}_{n\times 1} + \mathbf{D}_{m\times p}\mathbf{x}_{p\times 1} \end{cases} \tag{14}$$

Nonlinear Case

$$\begin{cases} \dot{\mathbf{s}}_{n\times 1} = \mathbf{f}_{n\times 1}(\mathbf{s}_{n\times 1}, \mathbf{x}_{p\times 1}) \\ \mathbf{y}_{m\times 1} = \mathbf{g}_{m\times 1}(\mathbf{s}_{n\times 1}, \mathbf{x}_{p\times 1}) \end{cases} \tag{15}$$

Canonical State-Space Representations

Controllable state-space forms

Controllability is the ability to drive the system states to arbitrary values through the input signal or noise in finite time (Kailath, 1980; Tsakalis, 2001)

Example

Controlling blood pressure through drug intake.

Observable state-space forms

Observability describes the ability to infer the system states given output measurements (Kailath, 1980; Tsakalis, 2001)

Example

Output demonstration of internal body infections as fever.

Question: Are physiological systems necessarily controllable and/or observable?

State Space Representations
(continued)

Example

$$a\frac{d^2y(t)}{dt^2} + b\frac{dy(t)}{dt} + cy(t) = x(t) \tag{16}$$

$$\begin{aligned} s_1 &\triangleq y(t) & \dot{s}_1 &= s_2 \\ s_2 &\triangleq \dot{y}(t) \Rightarrow & \dot{s}_2 &= \ddot{y}(t) = \tfrac{1}{a}(x(t) - bs_2 - cs_1) \end{aligned} \tag{17}$$

State-space representation:

$$\begin{aligned} \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -\frac{c}{a} & -\frac{b}{a} \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{a} \end{bmatrix} x(t) \\ y(t) &= [1 \quad 0] \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \end{aligned} \tag{18}$$

State-Space to Transfer Functions and Differential Equations

Taking the Laplace transform of (14):

$$\mathbf{Y}(s) = \left[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}\right]\mathbf{X}(s) \tag{19}$$

Transfer matrix: $\mathbf{H}(s) \triangleq \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$

- In contrast to state-space equations, $\mathbf{H}(s)$ is unique.
- Taking the inverse Laplace transform, takes the system back to the time domain (in differential equation or impulse response form)

Temporal Solutions from State-Space Representation

Scalar Case

Example

Consider the system $\dot{y}(t) = ay(t) + bx(t)$, with initial condition $y(0)$. Taking the Laplace transform, we find:

$$\begin{aligned} sY(s) - y(0) &= aY(s) + bX(s) \\ Y(s) &= \frac{y(0)}{s-a} + \frac{bX(s)}{s-a} \Rightarrow \\ y(t) &= y(0)e^{at}u(t) + be^{at}u(t) * x(t) \Rightarrow \\ y(t) &= y(0)e^{at}u(t) + b\int_0^t e^{a(t-\tau)}x(\tau)d\tau \end{aligned} \tag{20}$$

When $a < 0$, the effect of initial conditions vanish in time

Temporal Solutions from State-pace Representation

Vectorial Case

The vectorial differential equation $\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{B}\mathbf{x}(t)$ yields:

$$\begin{aligned} s\mathbf{Y}(s) - \mathbf{y}(0) &= \mathbf{A}\mathbf{Y}(s) + \mathbf{B}\mathbf{X}(s) \\ \mathbf{Y}(s) &= (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{y}(0) + (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{X}(s) \\ \mathbf{\Phi}(s) &\triangleq (s\mathbf{I} - \mathbf{A})^{-1} = L\left\{e^{\mathbf{A}t}\right\} = L\left\{\phi(t)\right\} \end{aligned} \tag{21}$$

where $\phi(t) = e^{\mathbf{A}t}u(t)$ is the state transition matrix.
Hence:

$$\mathbf{y}(t) = \phi(t)\mathbf{y}(0) + \int_0^t \phi(t-\tau)\mathbf{B}\mathbf{x}(\tau)d\tau \tag{22}$$

Or, if the initial states are given at $t = t_0$:

$$\mathbf{y}(t) = \phi(t-t_0)\mathbf{y}(t_0) + \int_{t_0}^t \phi(t-\tau)\mathbf{B}\mathbf{x}(\tau)d\tau \tag{23}$$

Whenever $\mathbf{x}(t) = \mathbf{0}$ we have: $\mathbf{y}(t) = \phi(t-t_0)\mathbf{y}(t_0)$

State Transition Matrix

Example

$$\dot{\mathbf{y}}(t) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathbf{y}(t) \tag{24}$$

with $\mathbf{y}(0) = \mathbf{0}$.
Solution:

$$\begin{aligned}
(s\mathbf{I} - \mathbf{A}) &= \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix} \\
(s\mathbf{I} - \mathbf{A})^{-1} &= \frac{1}{s^2 + 3s + 2} \begin{bmatrix} s+3 & 1 \\ -2 & s \end{bmatrix} \Rightarrow \\
\phi(t) = L^{-1}\{\Phi(s)\} &= \begin{bmatrix} -e^{-2t} + 2e^{-t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} u(t)
\end{aligned} \tag{25}$$

This matrix indicates variation of each state over time with respect to the specific initial condition.

State Transition Matrix
(continued)

Properties

- $\phi(0) = \mathbf{I}$: No state transition without time elapse (state transition requires time)
- $\phi^{-1}(\tau) = \phi(-\tau) \Rightarrow \mathbf{x}(t_0) = \phi^{-1}(t - t_0)\mathbf{x}(t)$: Time is reversible using the inverse transition matrix
- $\phi(t_1 + t_2) = \phi(t_1)\phi(t_2) = \phi(t_2)\phi(t_1)$: Time epochs can be split into smaller segments (special case: $[\phi(\Delta)]^k = \phi(k\Delta)$)
- $\phi(t_2 - t_0) = \phi(t_2 - t_1)\phi(t_1 - t_0)$: State transition is independent of the time path (like the work done in a conservative system)

Continuous vs. discrete models

- Models can be continuous or discrete in time.
- With current digital simulation and modeling systems, models are either discrete by definition or are discretized during the modeling and simulation procedure.
- Discretization in time consists of converting a continuous-time signal/system to a discrete-time signal/system.

Continuous vs. Discrete Models
(continued)

Examples of signal discretization and discrete signals

- **Discretized signal:** An ECG signal sampled and stored in a computer
- **Intrinsically discrete signal:** The heart rate time-series is discrete by definition

Examples of system discretization vs discrete systems

- **Discretized system:** Simulation of the respiratory system or cell-growth in a computer
- **Discrete systems:** Digital cardiac pacemakers, biofeedback systems, etc.

Signal Discretization

Signal discretization

- Signal discretization is the subject of *digital signal processing*
- The question is how to go from a continuous-time signal $x(t)$ to a discrete-time signal $x[n]$
- In the real world, discrete signals are achieved by sampling a continuous-time signal in a *uniform* or *non-uniform* manner.

System Discretization

Static system discretization

Since static systems do not evolve in time, the discretization is straightforward: $t \rightarrow nT_s$, where T_s is the sampling time.

Dynamic system discretization

Differential forms are approximated with difference equations of the appropriate order.

First-order discretization

For small Δ: $\frac{d}{dt}x(t) \approx \frac{x(t+\Delta)-x(t)}{\Delta}$. If T_s is small 'enough' (an order of magnitude smaller than the Nyquist frequency is typically fine), Δ can be replaced by T_s. Hence:

$$\frac{d}{dt}x(t) \rightarrow \frac{x[n+1]-x[n]}{T_s} \tag{26}$$

System Discretization

(continued)

- The same procedure can be used for state-space representations of models.
- For higher-order discretizations, the first-order can be extended or one may use an appropriate approximation of the desired order.

✎ The discretization error increases if T_s is not small enough.

System Discretization
(continued)

Discrete systems can be obtained from continuous ones using various approaches:

- Time-domain: The system is discretized from the *implicit* or *explicit* forms of its input/output equation
- Frequency-domain: $H(s)$ is transfered to $H(z)$ using a conformal mapping (cf. DSP textbooks)

✎ Discretized models are not unique

Static System Linearization

Consider a static system: $y = f(\mathbf{u})$ ($\mathbf{u} = [u_1, \cdots, u_n]^T$).

Static System Linearization

Consider a static system: $y = f(\mathbf{u})$ ($\mathbf{u} = [u_1, \cdots, u_n]^T$). A first-order Taylor expansion around $y_0 = f(\mathbf{u}_0)$ results in:

$$\begin{aligned} y &= f(\mathbf{u}_0) + \left.\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u_0}} (\mathbf{u} - \mathbf{u}_0) + \cdots \\ y &\approx f(\mathbf{u}_0) + \left.\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}}\right|_{\mathbf{u}=\mathbf{u_0}} (\mathbf{u} - \mathbf{u}_0) \end{aligned} \tag{27}$$

$$\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}} = \left.\left[\frac{\partial f(\mathbf{u})}{\partial u_1}, \cdots, \frac{\partial f(\mathbf{u})}{\partial u_n}\right]^T\right|_{\mathbf{u}=u_\mathbf{0}} \triangleq \mathbf{g}_0 \tag{28}$$

$$\begin{aligned} & y \approx f(\mathbf{u}_0) + \mathbf{g}_0^T(\mathbf{u} - \mathbf{u}_0) \Rightarrow \\ & y - y_0 \approx \mathbf{g}_0^T(\mathbf{u} - \mathbf{u}_0) \Rightarrow \\ & \delta y \approx \mathbf{g}_0^T \boldsymbol{\delta u} \end{aligned} \tag{29}$$

Static System Linearization
(continued)

Example

Linearize the following static system around $\bar{u}_1 = \bar{u}_2 = 0, \bar{y} = 0$:

$$y = u_1^2 u_2^2 + u_2 + 3u_1^2 \tag{30}$$

Static System Linearization
(continued)

Example

Linearize the following static system around $\bar{u}_1 = \bar{u}_2 = 0, \bar{y} = 0$:

$$y = u_1^2 u_2^2 + u_2 + 3u_1^2 \tag{30}$$

Approach 1 (Taylor expansion):

$$g^T = \begin{bmatrix} 2u_1 u_2^2 + 6u_1 & 2u_2 u_1^2 + 1 \end{bmatrix} \big|_{u_1=u_2=0} = [0 \quad 1] \Rightarrow \delta y \approx [0 \quad 1] \begin{bmatrix} \delta u_1 \\ \delta u_1 \end{bmatrix} = \delta u_2 \tag{31}$$

Static System Linearization
(continued)

Example

Linearize the following static system around $\bar{u}_1 = \bar{u}_2 = 0, \bar{y} = 0$:

$$y = u_1^2 u_2^2 + u_2 + 3u_1^2 \tag{30}$$

Approach 1 (Taylor expansion):

$$g^T = \begin{bmatrix} 2u_1 u_2^2 + 6u_1 & 2u_2 u_1^2 + 1 \end{bmatrix} |_{u_1=u_2=0} = [0 \quad 1] \Rightarrow \delta y \approx [0 \quad 1] \begin{bmatrix} \delta u_1 \\ \delta u_1 \end{bmatrix} = \delta u_2 \tag{31}$$

Approach 2 (Perturbation theory): Insert $u_1 = \bar{u}_1 + \delta u_1$, $u_2 = \bar{u}_2 + \delta u_2$, and $y = \bar{y} + \delta y$ in (30) and discard the second and higher-order terms:

$$(\bar{y} + \delta y) = (\bar{u}_1 + \delta u_1)^2 (\bar{u}_2 + \delta u_2)^2 + (\bar{u}_2 + \delta u_2) + 3(\bar{u}_1 + \delta u_1)^2 \tag{32}$$

which results in $\delta y \approx \delta u_2$ (*Note: this is only valid at* $\bar{u}_1 = \bar{u}_2 = 0$).

Dynamic System Linearization

The perturbation approach can also be used for linearizing dynamic systems:

Example

Linearize $y\dfrac{d^2y}{dt^2} + u^2\dfrac{dy}{dt} + \sqrt{y} = u$ around $\sqrt{y_0} = u_0$

Dynamic System Linearization

The perturbation approach can also be used for linearizing dynamic systems:

Example

Linearize $y\dfrac{d^2y}{dt^2} + u^2\dfrac{dy}{dt} + \sqrt{y} = u$ around $\sqrt{y_0} = u_0$

Solution: Replacing $u(t) = u_0 + \delta u(t)$ and $y(t) = y_0 + \delta y(t)$ gives:

$$(y_0 + \delta y)\frac{d^2}{dt^2}(y_0 + \delta y) + (u_0 + \delta u)^2\frac{d}{dt}(y_0 + \delta y) + \sqrt{y_0 + \delta y} = u_0 + \delta u \quad (33)$$

Replacing $\sqrt{y_0 + \delta y}$ with its Taylor expansion:

$$\sqrt{y_0 + \delta y} = (y_0 + \delta y)^{\frac{1}{2}} = y_0^{\frac{1}{2}} + \frac{1}{2\sqrt{y_0}}\delta y + \textit{(second and higher-order terms)} \quad (34)$$

Dynamic System Linearization
(continued)

Example (continued)

By discarding the second and higher-order terms, we find

$$y_0 \frac{d^2}{dt^2}(\delta y) + u_0^2 \frac{d}{dt}(\delta y) + \sqrt{y_0} + \frac{1}{2\sqrt{y_0}}\delta y \approx u_0 + \delta u \tag{35}$$

$$y_0 \delta \ddot{y} + u_0^2 \delta \dot{y} + \frac{1}{2\sqrt{y_0}}\delta y \approx \delta u \tag{36}$$

This approximation is only valid around (u_0, y_0). The transfer function (at this point) is:

$$H(s) = \frac{\Delta Y(s)}{\Delta U(s)} \approx \frac{1}{y_0 s^2 + u_0^2 s + \frac{1}{2\sqrt{y_0}}} \tag{37}$$

Depending on the application, $\delta u(t)$ and $\delta y(t)$ should be small enough to ensure the validity of these approximations.

Nonlinear State-Space System Linearization

Linearize the following system around $(\mathbf{u}_0, \mathbf{x}_0, \mathbf{y}_0)$:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \end{cases} \tag{38}$$

The overall approach is similar:

$$\begin{cases} \mathbf{u}(t) = \mathbf{u}_0 + \delta\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{y}_0 + \delta\mathbf{y}(t) \\ \mathbf{x}(t) = \mathbf{x}_0 + \delta\mathbf{x}(t) \end{cases} \tag{39}$$

$$\begin{aligned} \dot{\mathbf{x}}_0 + \delta\dot{\mathbf{x}}(t) = \quad & \mathbf{f}(\mathbf{x}_0 + \delta\mathbf{x}(t), \mathbf{u}_0 + \delta\mathbf{u}(t)) \\ & = \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{y}_0}} \delta\mathbf{x}(t) + \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\bigg|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{y}_0}} \delta\mathbf{u}(t) \\ & + \cdots \\ \mathbf{y}_0 + \delta\mathbf{y}(t) = \quad & \mathbf{g}(\mathbf{x}_0 + \delta\mathbf{x}(t), \mathbf{u}_0 + \delta\mathbf{u}(t)) \\ & = \mathbf{g}(\mathbf{x}_0, \mathbf{u}_0) + \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{y}_0}} \delta\mathbf{x}(t) + \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}}\bigg|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{y} = \mathbf{y}_0}} \delta\mathbf{u}(t) \\ & + \cdots \end{aligned} \tag{40}$$

Nonlinear State-Space System Linearization
(continued)

Defining

$$
\mathbf{A}(t) \triangleq \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{y}=\mathbf{y}_0}} \qquad \mathbf{B}(t) \triangleq \left.\frac{\partial \mathbf{f}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{y}=\mathbf{y}_0}}
$$
$$
\mathbf{C}(t) \triangleq \left.\frac{\partial \mathbf{g}(\mathbf{x},\mathbf{u})}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{y}=\mathbf{y}_0}} \qquad \mathbf{D}(t) \triangleq \left.\frac{\partial \mathbf{g}(\mathbf{x},\mathbf{u})}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\mathbf{x}_0\\ \mathbf{y}=\mathbf{y}_0}} \tag{41}
$$

and considering that

$$
\begin{cases} \dot{\mathbf{x}_0} = \mathbf{f}(\mathbf{x}_0,\mathbf{u}_0) \\ \mathbf{y}_0 = \mathbf{g}(\mathbf{x}_0,\mathbf{u}_0) \end{cases} \tag{42}
$$

Results in:

$$
\begin{cases} \delta\dot{\mathbf{x}}(t) \approx \mathbf{A}(t)\delta\mathbf{x}(t) + \mathbf{B}(t)\delta\mathbf{u}(t) \\ \delta\mathbf{y}(t) \approx \mathbf{C}(t)\delta\mathbf{x}(t) + \mathbf{D}(t)\delta\mathbf{u}(t) \end{cases} \tag{43}
$$

which is a linear approximation of the original nonlinear state-space equations around $(\mathbf{u}_0, \mathbf{x}_0, \mathbf{y}_0)$.

Stability Analysis of Nonlinear Systems

In linear systems

All the system poles should be in the left half plane

In nonlinear systems

1. The system is first linearized around the desired point
2. The state transition matrix is calculated for the linearized system: $\phi(s) = (s\boldsymbol{I} - \boldsymbol{A})^{-1}$ $(\phi(t) = L^{-1}\{\phi(s)\})$
3. In order to have a stable system, the poles of $\phi(s)$ should be in the left half plane (or $\lim_{t\to\infty} \phi(t) = 0$), around the point of interest

Further Reading

- Kailath, T. (1980). *Linear Systems.*
  Prentice Hall
- Dorf, R. C. & Bishop, R. H. (2021). *Modern Control Systems.*
  Pearson Prentice Hall, 13th edition
- Ogata, K. (2010). *Modern control engineering.*
  Prentice hall, 5th edition
- Oppenheim, A., Schafer, R., & Buck, J. (1999). *Discrete-time signal processing.*
  Prentice-Hall signal processing series. Prentice Hall
- Tsakalis, K. S. (2001). Stability, controllability, observability.
  Lecture Notes