Linear Time-Invariant Dynamical Systems

CEE 629. System Identification
Duke University

Henri P. Gavin
Spring 2019

1 Linearity and Time Invariance

A system $\mathcal{G}$ that maps an input $u(t)$ to an output $y(t)$ is a *linear system* if and only if

$$(\alpha_1 y_1(t) + \alpha_2 y_2(t)) = \mathcal{G}[\alpha_1 u_1(t) + \alpha_2 u_2(t)] \tag{1}$$

where $y_1 = \mathcal{G}[u_1]$, $y_2 = \mathcal{G}[u_2]$ and α_1 and α_2 are scalar constants. If (1) holds only for $\alpha_1 + \alpha_2 = 1$ the system is called *affine.* The function $\mathcal{G}(u) = Au + b$ is affine, but not linear.

A system $\mathcal{G}$ that maps an input $u(t)$ to an output $y(t)$ is a *time-invariant system* if and only if

$$y(t - t_o) = \mathcal{G}[u(t - t_o)] \ . \tag{2}$$

Systems described by

$$\dot{x}(t) = Ax(t) + Bu(t) \ , \qquad x(0) = x_o \tag{3}$$

$$y(t) = Cx(t) + Du(t) \tag{4}$$

are linear and time-invariant.

variable	**description**	**dimension**
x	state vector	n by 1
u	input vector	r by 1
y	output vector	m by 1
A	dynamics matrix	n by n
B	input matrix	n by r
C	output matrix	m by n
D	feedthrough matrix	m by r

2 Example: a spring-mass-damper oscillator

An externally-forced spring-mass-damper oscillator is described by

$$m\ddot{r}(t) + c\dot{r}(t) + kr(t) = f(t) \ , \qquad r(0) = d_o \ , \quad \dot{r}(0) = v_o \ . \tag{5}$$

Setting the external forcing to zero, substituting an assumed solution of the form $r(t) = \bar{r}e^{\lambda t}$, and factoring out the $e^{\lambda t}$, results in

$$(m\lambda^2 + c\lambda + k)\bar{r} = 0 \ .$$

This equation is valid for $\bar{r} = 0$ (the trivial solution) and for $(m\lambda^2 + c\lambda + k) = 0$, which is called the *characteristic equation* of this differential equation. The roots of this polynomial are given by the quadratic formula,

$$\lambda = -\frac{c}{2m} \pm \sqrt{\frac{c^2}{4m} - \frac{k}{m}} \ .$$

Defining the *natural frequency* $\omega_{\rm n}^2 \equiv k/m$ and the *damping ratio* $\zeta \equiv c/(2\sqrt{mk})$, we find $c/(2m) = \zeta\omega_{\rm n}$, so,

$$\begin{aligned} \lambda &= -\zeta\omega_{\rm n} \pm \sqrt{\zeta^2\omega_{\rm n}^2 - \omega_{\rm n}^2} \\ &= -\zeta\omega_{\rm n} \pm \omega_{\rm n}\sqrt{\zeta^2 - 1} \end{aligned} \tag{6}$$

and if $\zeta < 1$, the root may be written

$$\lambda = -\zeta\omega_{\rm n} \pm {\rm i}\omega_{\rm n}\sqrt{1-\zeta^2} \qquad \left({\rm i} = \sqrt{-1}\right) .$$

Complex values of λ are written $\lambda = \sigma \pm {\rm i}\omega$. Note that $(\zeta > 0) \Leftrightarrow (c > 0) \Leftrightarrow (\sigma < 0) \Leftrightarrow$ the simple oscillator is stable.

Now presuming that the external forcing $f(t)$ and the position response $r(t)$ are harmonic, $f(t) = \bar{f}(s)e^{st}$ and $r(t) = \bar{r}(s)e^{st}$ $(s \in \mathbb{C})$, substituting the presumed solution into the differential equation and factoring out e^{st}

$$(ms^2 + cs + k)\ \bar{r}(s) = \bar{f}(s) \qquad \text{or} \qquad \bar{r}(s) = \bar{f}(s)\ /\ (ms^2 + cs + k)\ .$$

Now, considering inputs $u(t) = f(t)$ and outputs $y_1(t) = \ddot{r}(t)$ and $y_2(t) = kr(t) + c\dot{r}(t)$ and their Laplace transforms, $y_1(t) = \bar{y}_1(s)e^{st} = s^2\bar{r}(s)e^{st}$ and

$y_2(t) = \bar{y}_2(s)e^{st} = k\bar{r}(s)e^{st} + cs\bar{r}(s)e^{st}$ we can derive transfer functions from $\bar{u}(s)$ to $\bar{y}_1(s)$ and $\bar{y}_2(s)$.

$$\frac{\bar{y}_1(s)}{\bar{u}(s)} = \frac{s^2}{ms^2 + cs + k} \tag{7}$$

and

$$\frac{\bar{y}_2(s)}{\bar{u}(s)} = \frac{k + cs}{ms^2 + cs + k} \tag{8}$$

The second-order ordinary differential equation (5) may be written as two first-order ordinary differential equations, by defining a *state vector* of the position and velocity, $x = [r \ \dot{r}]^{\mathsf{T}}$.

$$\frac{d}{dt}\begin{bmatrix} r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} f(t) \ , \qquad \begin{bmatrix} r(0) \\ \dot{r}(0) \end{bmatrix} = \begin{bmatrix} d_o \\ v_o \end{bmatrix} \tag{9}$$

In terms of a desired response from this system, we may be interested in the force on the foundation, f_{F}, and the acceleration of the mass, both of which can be computed directly through a linear combination of the states and the input.

$$\begin{bmatrix} f_{\mathrm{F}} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} k & c \\ -k/m & -c/m \end{bmatrix} \begin{bmatrix} r \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} f(t) \tag{10}$$

A single degree of freedom oscillator and all other linear dynamical systems may be described in a general sense using a state variable realization,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \ , \qquad x(0) = x_o \\ y(t) &= Cx(t) + Du(t) \ . \end{aligned}$$

The next section shows the equivalence of differential equations, transfer functions, and state variable realizations, and shows how the state variable realizations (9) and (10) can be obtained directly from the transfer functions (7) and (8) without considering the differential equations.

3 System Interconnections: parallel, cascade, and feedback

The facility with which models of interconnected subsystems can be derived is one of the powerful benefits of state-space modeling. This section describes the three fundamental types of system interconnections: parallel, cascade, and feedback. The individual interconnected subsystems are described by:

$$\dot{x}_1 = A_1 x_1 + B_1 u_1 \; , \qquad y_1 = C_1 x_1 + D_1 u_1$$

$$\dot{x}_2 = A_2 x_2 + B_2 u_2 \; , \qquad y_2 = C_2 x_2 + D_2 u_2$$

3.1 Parallel interconnections

In the parallel interconnection of two subsystems, the same output drives both subsystems, $u_1 = u_2 = u$, and the output is the sum of the two subsytem outputs, $y = y_1 + y_2$. So,

$$y = C_1 x_1 + D_1 u + C_2 x_2 + D_2 u$$

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \tag{11}$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_1 + D_2 \end{bmatrix} u \tag{12}$$

3.2 Cascade interconnections

In the cascade interconnection of two subsystems, the output of subsystem 1 provides the input to subsystem 2, $u_2 = y_1$. So,

$$\dot{x}_2 = A_2 x_2 + B_2(C_1 x_1 + D_1 u_1) \qquad y_2 = C_2 x_2 + D_2(C_1 x_1 + D_1 u_1)$$

$$\dot{x}_2 = B_2 C_1 x_1 + A_2 x_2 + B_2 D_1 u_1 \qquad y_2 = D_2 C_1 x_1 + C_2 x_2 + D_2 D_1 u_1$$

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ B_2 C_1 & A_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_1 + B_2 D_1 \end{bmatrix} u_1 \tag{13}$$

$$y_2 = \begin{bmatrix} D_2 C_1 & C_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_2 D_1 \end{bmatrix} u_1 \tag{14}$$

3.3 Feedback interconnections

In the feedback interconnection of two subsystems, the output of subsystem 1 provides the input to subsystem 2, and the input to subsystem 1 is the sum of the output of subsystem 2 and the overall system input, u. The overall system output is the output of subsystem 1.

$$u_2 = y_1 \qquad \text{and} \qquad u_1 = u + y_2 \quad \text{and} \qquad y = y_1$$

So,

$$\dot{x}_1 = A_1 x_1 + B_1(u + y_2) \qquad y = C_1 x_1 + D_1(u + y_2)$$

$$\dot{x}_2 = A_2 x_2 + B_2(C_1 x_1 + D_1(u + y_2)) \qquad y_2 = C_2 x_2 + D_2(C_1 x_1 + D_1(u + y_2))$$

$$y_2 = (I - D_1)^{-1} D_2 C_1 x_1 + (I - D_1)^{-1} C_2 x_2 + (I - D_1)^{-1} D_1 u$$

$$\dot{x}_1 = A_1 x_1 + B_1 (I - D_1)^{-1} D_2 C_1 x_1 + B_1 (I - D_1)^{-1} C_2 x_2 + B_1 (I - D_1)^{-1} D_1 u + B_1 u$$

$$\dot{x}_1 = (A_1 + B_1 (I - D_1)^{-1} D_2 C_1) x_1 + B_1 (I - D_1)^{-1} C_2 x_2 + B_1 ((I - D_1)^{-1} D_1 + I) u$$

$$\dot{x}_2 = B_2 C_1 x_1 + A_2 x_2 + B_2 D_1 y_2 + B_2 D_1 u$$

$$\dot{x}_2 = B_2 C_1 x_1 + A_2 x_2 + B_2 D_1 ((I - D_1)^{-1} D_2 C_1 x_1 + (I - D_1)^{-1} C_2 x_2 + (I - D_1)^{-1} D_1 u) + B_2 D_1 u$$

$$\dot{x}_2 = (B_2 C_1 + B_2 D_1 (I - D_1)^{-1} D_2 C_1) x_1 + (A_2 + B_2 D_1 (I - D_1)^{-1} C_2) x_2 + B_2 D_1 ((I - D_1)^{-1} D_1 + I) u$$

$$y = C_1 x_1 + D_1 ((I - D_1)^{-1} D_2 C_1 x_1 + (I - D_1)^{-1} C_2 x_2 + (I - D_1)^{-1} D_1 u) + D_1 u$$

$$y = (C_1 + D_1 (I - D_1)^{-1} D_2 C_1) x_1 + D_1 (I - D_1)^{-1} C_2 x_2 + D_1 ((I - D_1)^{-1} + I) D_1 u$$

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} A_1 + B_1 (I - D_1)^{-1} D_2 C_1 & B_1 (I - D_1)^{-1} C_2 \\ B_2 C_1 + B_2 D_1 (I - D_1)^{-1} D_2 C_1 & A_2 + B_2 D_1 (I - D_1)^{-1} C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 ((I - D_1)^{-1} D_1 + I) \\ B_2 D_1 ((I - D_1)^{-1} D_1 + I) \end{bmatrix} u \\ y &= \begin{bmatrix} C_1 + D_1 (I - D_1)^{-1} D_2 C_1 & D_1 (I - D_1)^{-1} C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} D_1 ((I - D_1)^{-1} + I) D_1 \end{bmatrix} u \end{aligned}$$

In the special case where $D_1 = 0$ and $D_2 = 0$,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_1 & B_1 C_2 \\ B_2 C_1 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u \tag{15}$$

$$y = \begin{bmatrix} C_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u \tag{16}$$

4 Differential Equations, Transfer Functions, and Continuous Time State Space Realizations

In general, any linear ordinary differential equation with constant coefficients

$$\begin{aligned} & a_0 y(t) + a_1 \dot{y}(t) + a_2 \ddot{y}(t) + \cdots + a_{n-1} y^{(n-1)}(t) + y^{(n)}(t) \\ = \; & b_0 u(t) + b_1 \dot{u}(t) + b_2 \ddot{u}(t) + \cdots + b_{n-1} u^{(n-1)}(t) + b_n u^{(n)}(t) \end{aligned} \tag{17}$$

can be expressed in state-space form as long as the highest order of the derivitives of u do not exceed the highest order of the derivitives of y. Setting the external forcing, $u(t)$ and all its derivitives, to zero and substituting an assumed solution of the form $y(t) = \bar{y}e^{\lambda t}$, and factoring out the $e^{\lambda t}$, results in

$$(\; a_0 + a_1\lambda + a_2\lambda^2 + \cdots + a_{n-1}\lambda^{n-1} + \lambda^n \;)\; \bar{y} = 0.$$

This equation is valid for $\bar{y} = 0$ (the trivial solution) and for

$$a_o + a_1\lambda + a_2\lambda^2 + \cdots + a_{n-1}\lambda^{n-1} + \lambda^n \;=\; 0 \;, \tag{18}$$

which is the *characteristic equation* of the differential equation (17). For $n > 3$ the n roots of this polynomial, $(\lambda_1, ..., \lambda_n)$ may be computed numerically. In general, these roots are complex and are conventionally expressed as

$$\lambda_i = \sigma_i \pm \mathrm{i}\omega_i \;, \qquad \left(\mathrm{i} = \sqrt{-1}\right).$$

Now considering harmonically forced steady state inputs and outputs, assume a harmonic input of the form $u(t) = \bar{u}(s)e^{st}$ and a harmonic output of the form $y(t) = \bar{y}(s)e^{st}$. Allowing the *Laplace variable* to be complex, $s \in \mathbb{C}$, these assumed solutions can represent both harmonic and exponential functions. Substituting the assumed solutions into the differential equation, and factoring out e^{st} from both sides, gives the differential equation expressed in the Laplace domain.

$$\begin{aligned} & (\; a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1}s^{n-1} + s^n \;)\; \bar{y}(s) \\ = \; & (\; b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1}s^{n-1} + b_n s^n \;)\; \bar{u}(s) \end{aligned}$$

The ratio of the output $\bar{y}(s)$ to the input $\bar{u}(s)$ in the Laplace domain is called the *transfer function*

$$H(s) \equiv \frac{\bar{y}(s)}{\bar{u}(s)} = \frac{b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1} s^{n-1} + b_n s^n}{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1} + s^n} \tag{19}$$

To obtain a state space realization of this differential equation, we convert the Laplace domain transfer function back to a time domain differential equation by multiplying the numerator and the denominator of the transfer function by the same Laplace domain variable $\bar{v}(s)$, which will be used to represent the states of the system.

$$\frac{\bar{y}(s)}{\bar{u}(s)} = \frac{(\ b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1} s^{n-1} + b_n s^n\ )\ \bar{v}(s)}{(\ a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1} + s^n\ )\ \bar{v}(s)} \tag{20}$$

Now defining

$$\begin{aligned} \bar{y}(s) &\equiv (\ b_0 + b_1 s + b_2 s^2 + \cdots + b_{n-1} s^{n-1} + b_n s^n\ )\ \bar{v}(s)\ , \\ \bar{u}(s) &\equiv (\ a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1} + s^n\ )\ \bar{v}(s)\ , \end{aligned}$$

taking the inverse Laplace transform,

$$\begin{aligned} y(t) &= b_0 v(t) + b_1 \dot{v}(t) + b_2 \ddot{v}(t) + \cdots + b_{n-1} v^{(n-1)}(t) + b_n v^{(n)}(t)\ , \\ u(t) &= a_0 v(t) + a_1 \dot{v}(t) + a_2 \ddot{v}(t) + \cdots + a_{n-1} v^{(n-1)}(t) + v^{(n)}(t)\ , \end{aligned}$$

and defining states

$$\begin{array}{ccccc} x_1 &=& v(t) & & \\ x_2 &=& \dot{x}_1 &=& \dot{v}(t)\ , \\ x_3 &=& \dot{x}_2 &=& \ddot{v}(t)\ , \\ \vdots & & \vdots & & \vdots \\ x_n &=& \dot{x}_{n-1} &=& v^{(n-1)}(t)\ , \\ \dot{x}_n &=& f(x_1, ..., x_n, u) &=& v^{(n)}(t)\ , \end{array}$$

we obtain expressions for $y(t)$ and $u(t)$ in terms of n states x_1, ... , x_n, and $\dot{x}_n$.

$$\begin{aligned} y(t) &= b_0 x_1(t) + b_1 x_2(t) + b_2 x_3(t) + \cdots + b_{n-1} x_n(t) + b_n \dot{x}_n(t) \\ u(t) &= a_0 x_1(t) + a_1 x_2(t) + a_2 x_3(t) + \cdots + a_{n-1} x_n(t) + \dot{x}_n(t) \end{aligned}$$

Solving the second equation for $\dot{x}_n(t)$ we obtain the highest state derivitive as a function of the states and the input

$$\dot{x}_n(t) = u(t) - a_0 x_1(t) - a_1 x_2(t) - a_2 x_3(t) - \cdots - a_{n-1} x_n(t) \ .$$

Inserting this equation into the equation for $y(t)$, we obtain the output equation as a function of states and the input,

$$\begin{aligned} y(t) &= b_0 x_1(t) + b_1 x_2(t) + b_2 x_3(t) + \cdots + b_{n-1} x_n(t) \\ &+ b_n(\ u(t) - a_0 x_1(t) - a_1 x_2(t) - a_2 x_3(t) - \cdots - a_{n-1} x_n(t)\ ) \ . \end{aligned}$$

Combining terms with the same states

$$\begin{aligned} y(t) &= (b_0 - a_0 b_n)\ x_1(t) + (b_1 - a_1 b_n)\ x_2(t) + (b_2 - a_2 b_n)\ x_3(t) + \cdots \\ &+ (b_{n-1} - a_{n-1} b_n)\ x_n(t) + b_n u(t) \end{aligned}$$

and combining with the definition of the states, leads to a system of first order linear differential equations for the single n-th order ordinary differential equation.

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) \tag{21}$$

$$y(t) = \begin{bmatrix} b_0 - a_0 b_n & b_1 - a_1 b_n & b_2 - a_2 b_n & \cdots & b_{n-1} - a_{n-1} b_n \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} + [b_n]\ u(t)$$

This is called the *controllable canonical companion matrix* state space realization of the system described by differential equation (17) or transfer function (19). The coefficients $a_0, \cdots, a_{n-1}$ or, equivalently, the roots $\lambda_1, \cdots, \lambda_n$, of the characteristic equation, and the input coefficients $b_0, ..., b_n$ fully specify the system.

The dynamics matrix A of the canonical controlable companion matrix realization, can be recovered by its n eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$, provided the eigenvalues are distinct, by noting that for any eigenvalue λ_j,

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ \lambda_j \\ \lambda_j^2 \\ \vdots \\ \lambda_j^{n-1} \end{bmatrix} = \lambda_j \begin{bmatrix} 1 \\ \lambda_j \\ \vdots \\ \lambda_j^{n-2} \\ \lambda_j^{n-1} \end{bmatrix}$$

in which the last row is the characteristic equation (18). The Vandermonde matrix built from columns

$$\bar{V} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_n \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_n^2 \\ \vdots & \vdots & & \vdots \\ \lambda_1^{n-1} & \lambda_2^{n-1} & \cdots & \lambda_n^{n-1} \end{bmatrix}$$

contain the eigenvectors of the dynamics matrix of the conrollable canonical companion matrix realization, so $A\bar{V} = \bar{V}\Lambda$ and $A = \bar{V}\Lambda\bar{V}^{-1}$, where $\Lambda = \mathsf{diag}(\lambda_1, ...\lambda_n)$ and provided $\bar{V}$ is full rank.

A somewhat more cumbersome approach leads to the *observable canonical companion matrix* state space realization of a differential equation or transfer function. To do so, we solve the differential equation (17) for $y^{(n)}(t)$ and integrate this equation $(n-1)$ times to get an expression for $\dot{y}(t)$, and then collect terms with common orders of integration or differentiation,

$$\begin{aligned}
y^{(n)}(t) &= -a_0 y(t) - a_1 \dot{y}(t) - a_2 \ddot{y}(t) - \cdots - a_{n-1} y^{(n-1)}(t) \\
&+ b_0 u(t) + b_1 \dot{u}(t) + b_2 \ddot{u}(t) + \cdots + b_{n-1} u^{(n-1)}(t) + b_n u^{(n)}(t) \\
\dot{y}(t) &= -a_0 \int_{n-1} y(t) dt^{n-1} - a_1 \int_{n-2} y(t) dt^{n-2} - a_2 \int_{n-3} y(t) dt^{n-3} - \cdots \\
&- a_{n-1} y(t) \\
&+ b_0 \int_{n-1} u(t) dt^{n-1} + b_1 \int_{n-2} u(t) dt^{n-2} + b_2 \int_{n-3} u(t) dt^{n-3} + \cdots \\
&+ b_{n-1} u(t) + b_n \dot{u}(t) \\
\dot{y}(t) - b_n \dot{u}(t) &= \int_{n-1} b_0 u(t) - a_0 y(t) dt^{n-1} + \int_{n-2} b_1 u(t) - a_1 y(t) dt^{n-2} \\
&+ \int_{n-3} b_2 u(t) - a_2 y(t) dt^{n-3} + \cdots + b_{n-1} u(t) - a_{n-1} y(t)
\end{aligned}$$

In these equations $\int_p f(t) dt^p$ is shorthand for integrating $f(t)$ p times. Now we define the first state to be the integral of the left hand side of the last expression and the second state to be all the integrals on the right hand side

$$\begin{aligned}
x_1(t) &\equiv y(t) - b_n u(t) \\
x_2(t) &\equiv \int_{n-1} b_0 u(t) - a_0 y(t) dt^{n-1} + \int_{n-2} b_1 u(t) - a_1 y(t) dt^{n-2} \\
&+ \int_{n-3} b_2 u(t) - a_2 y(t) dt^{n-3} + \cdots + \int b_{n-2} u(t) - a_{n-2} y(t) dt
\end{aligned} \tag{22}$$

giving us the first state equation

$$\begin{aligned}
\dot{x}_1 &= x_2 + b_{n-1} u(t) - a_{n-1} y(t) \\
&= -a_{n-1} x_1(t) + x_2 + b_{n-1} u(t) - a_{n-1} b_n u(t)
\end{aligned}$$

The derivitive of x_2 is

$$\begin{aligned}
\dot{x}_2(t) &= \int_{n-2} b_0 u(t) - a_0 y(t) dt^{n-2} + \int_{n-3} b_1 u(t) - a_1 y(t) dt^{n-3} \\
&+ \int_{n-4} b_2 u(t) - a_2 y(t) dt^{n-4} + \cdots + b_{n-2} u(t) - a_{n-2} y(t)
\end{aligned}$$

Defining the terms with integrals in the expression above as $x_3(t)$

$$\begin{aligned}
x_3(t) &\equiv \int_{n-2} b_0 u(t) - a_0 y(t) dt^{n-2} + \int_{n-3} b_1 u(t) - a_1 y(t) dt^{n-3} \\
&+ \int_{n-4} b_2 u(t) - a_2 y(t) dt^{n-4} + \cdots + \int b_{n-3} u(t) - a_{n-3} y(t) dt
\end{aligned}$$

gives us the second state equation

$$\begin{aligned}\dot{x}_2 &= x_3 + b_{n-2}u(t) - a_{n-2}y(t) \\ &= -a_{n-2}x_1(t) + x_3 + b_{n-2}u(t) - a_{n-2}b_n u(t)\end{aligned}$$

One more time — the derivitive of x_3 is

$$\begin{aligned}\dot{x}_3(t) &= \int_{n-3} b_0 u(t) - a_0 y(t) dt^{n-3} + \int_{n-4} b_1 u(t) - a_1 y(t) dt^{n-4} \\ &+ \int_{n-5} b_2 u(t) - a_2 y(t) dt^{n-5} + \cdots + b_{n-3}u(t) - a_{n-3}y(t)\end{aligned}$$

Defining the terms with integrals in the expression above as $x_4(t)$

$$\begin{aligned}x_4(t) &\equiv \int_{n-3} b_0 u(t) - a_0 y(t) dt^{n-3} + \int_{n-4} b_1 u(t) - a_1 y(t) dt^{n-4} \\ &+ \int_{n-5} b_2 u(t) - a_2 y(t) dt^{n-5} + \cdots + b_{n-3}u(t) - a_{n-3}y(t)\end{aligned}$$

gives us the third state equation

$$\begin{aligned}\dot{x}_3 &= x_4 + b_{n-3}u(t) - a_{n-3}y(t) \\ &= -a_{n-3}x_1(t) + x_4 + b_{n-3}u(t) - a_{n-3}b_n u(t)\end{aligned}$$

Repeating the pattern, we obtain the observable canonical companion matrix realization.

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} = \begin{bmatrix} -a_{n-1} & 1 & 0 & \cdots & 0 \\ -a_{n-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -a_1 & 0 & 0 & \cdots & 1 \\ -a_0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} b_{n-1} - a_{n-1}b_n \\ b_{n-2} - a_{n-2}b_n \\ \vdots \\ b_1 - a_1 b_n \\ b_0 - a_0 b_n \end{bmatrix} u(t) \tag{23}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_n(t) \end{bmatrix} + [b_n]\, u(t)$$

Notes:

- State space realizations for a given differential equation or transfer function are not unique.

- The observable canonical realization is the flipped transpose of the controllable canonical realization. By reversing the order of the states, the A matrix for the observable canonical realization is the transpose of A for the controllable canonical realization; B for the observable canonical realization is the transpose of C for the controllable canonical realization; and transpose of C for the observable canonical realization is B for the controllable canonical realization. The feedthrough matrix D is realization independent, as it should be.

- The roots of a characteristic equations of an ordinary differential equation with real valued coefficients are real or occur in complex conjugate pairs. In the matlab language, the vector $[a_0,\ a_1,\ a_2,\ \cdots,\ a_{n-1},\ 1]$ is encoded in reverse (“`fliped`”) order as as $[1,\ a_{n-1},\ a_{n-2},\ \cdots,\ a_1,\ a_0]$. So computations can be carried out as,

```
a = flip(real(poly(lambda)));                 % coeff’s from eigenvalue
A = [ zeros(N-1,1) , eye(N-1) ; -a(1:N) ]; % dynamics matrix
lambda = roots(flip(a));                      % eigenvalues from coeff’
```

5 Difference Equations, Transfer Functions, and Discrete Time State Space Realizations

In discrete time where variables are sampled at uniform time increments (Δt), $y(k)$ is shorthand for $y(k\Delta t)$, and $u(k)$ is shorthand for $u(k\Delta t)$. Any linear ordinary difference equation with constant coefficients

$$\begin{aligned} & y(k) + a_1 y(k-1) + a_2 y(k-2) + \cdots + a_{n-1} y(k-n+1) + a_n y(k-n) \\ = \; & b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_{n-1} u(k-n+1) + b_n u(k-n) \end{aligned} \tag{24}$$

can be expressed in state-space form as long as the highest time lag of u does not exceed the longest time lag of y. Setting the external forcing, $u(k)$ to zero for all k, and substituting an assumed solution of the form $y(n) = \bar{y} e^{\lambda k \Delta t}$, and factoring out the $e^{\lambda k \Delta t}$, results in

$$(\; 1 + a_1 \lambda^{-1} + a_2 \lambda^{-2} + \cdots + a_{n-1} \lambda^{-n+1} + a_n \lambda^{-n} \;) \; \bar{y} = 0.$$

This equation is valid for $\bar{y} = 0$ (the trivial solution) and for $(a_o + a_1 \lambda^{-1} + \cdots + \lambda^{-n}) \; = \; 0$, which is called the *characteristic equation* of this difference equation. For $n > 3$ the n roots of this polynomial, $(\lambda_1, ..., \lambda_n)$ may be computed numerically. In general, these roots are complex and are conventionally expressed as

$$\lambda_i = \sigma_i \pm \mathrm{i}\omega_i \; , \qquad \left(\mathrm{i} = \sqrt{-1}\right).$$

Now considering harmonically forced steady state inputs and outputs, assume a harmonic input of the form $u(t) = \bar{u}(s) e^{sk\Delta t} \equiv \bar{u}(z) z^k$ and a harmonic output of the form $y(t) = \bar{y}(s) e^{sk\Delta t} \equiv \bar{y}(z) z^k$. Allowing the *Laplace variable* to be complex, $s \in \mathbb{C}$, these assumed solutions can represent both harmonic and exponential functions. Substituting the assumed solutions into the difference equation, and factoring out $e^{sk\Delta t}$ from both sides, gives the difference equation expressed in the z-domain.

$$\begin{aligned} & (\; 1 \; + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n} \;) \; \bar{y}(z) \\ = \; & (\; b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n} \;) \; \bar{u}(z) \end{aligned}$$

The ratio of the output $\bar{y}(z)$ to the input $\bar{u}(z)$ in the z-domain is called the *transfer function*

$$H(z) \equiv \frac{\bar{y}(z)}{\bar{u}(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n}}{1 \ + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}} \tag{25}$$

To obtain a state space realization of this difference equation, we convert the z-domain transfer function back to a time domain difference equation by multiplying the numerator and the denominator of the transfer function by the same z-domain variable $\bar{v}(z)$, which will be used to represent the states of the system.

$$\frac{\bar{y}(z)}{\bar{u}(z)} = \frac{(\ b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n}\ )\ \bar{v}(z)}{(\ 1 \ + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}\ )\ \bar{v}(z)} \tag{26}$$

Now defining

$$\begin{aligned}
\bar{y}(z) &\equiv (\ b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n-1} z^{-n+1} + b_n z^{-n}\ )\ \bar{v}(z)\ , \\
\bar{u}(z) &\equiv (\ 1 \ + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n-1} z^{-n+1} + a_n z^{-n}\ )\ \bar{v}(z)\ ,
\end{aligned}$$

taking the inverse z-transform,

$$\begin{aligned}
y(k) &= b_0 v(k) + b_1 v(k-1) + b_2 v(k-2) + \cdots + b_{n-1} v(k-n+1) + b_n v(k-n)\ , \\
u(k) &= v(k) + a_1 v(k-1) + a_2 v(k-2) + \cdots + a_{n-1} v(k-n+1) + a_n v(k-n)\ ,
\end{aligned}$$

and defining states

$$\begin{array}{ccccc}
x_1(k) & = & v(k-n) & & \\
x_2(k) & = & x_1(k+1) & = & v(k-n+1)\ , \\
x_3(k) & = & x_2(k+1) & = & v(k-n+2)\ , \\
\vdots & & \vdots & & \vdots \\
x_n(k) & = & x_{n-1}(k+1) & = & v(k-1)\ , \\
x_n(k+1) & = & f(x_1, ..., x_n, u) & = & v(k)\ ,
\end{array}$$

we obtain expressions for $y(k)$ and $u(k)$ in terms of n states x_1, ... , x_n, and $x_n(k+1)$.

$$\begin{aligned}
y(k) &= b_0 x_n(k+1) + b_1 x_n(k) + b_2 x_{n-1}(k) + \cdots + b_{n-1} x_2(k) + b_n x_1(k) \\
u(k) &= x_n(k+1) + a_1 x_n(k) + a_2 x_{n-1}(k) + \cdots + a_{n-1} x_2(k) + a_n x_1(k)
\end{aligned}$$

Solving the second equation for $x_n(k+1)$ we obtain the highest state difference as a function of the states and the input

$$x_n(k+1) = u(k) - a_1 x_n(k) - a_2 x_{n-1}(k) - \cdots - a_{n-1} x_2(k) - a_n x_1(k) \ .$$

Inserting this equation into the equation for $y(k)$, we obtain the output equation as a function of states and the input,

$$\begin{aligned} y(k) &= b_0 \left(u(k) - a_1 x_n(k) - a_2 x_{n-1}(k) - \cdots - a_{n-1} x_2(k) - a_n x_1(k)\right) \\ &+ b_1 x_n(k) + b_2 x_{n-1}(k) + \cdots + b_{n-1} x_2(k) + b_n x_1(k) \end{aligned}$$

Combining terms with the same states

$$\begin{aligned} y(k) &= (b_n - a_n b_0)\ x_1(k) + (b_{n-1} - a_{n-1} b_0)\ x_2(k) + (b_{n-2} - a_{n-2} b_0)\ x_3(k) + \cdots \\ &+ (b_2 - a_2 b_0)\ x_{n-1}(k) + (b_1 - a_1 b_0)\ x_n(k) + b_0\ u(k) \end{aligned}$$

and combining with the definition of the states, leads to a system of first order linear difference equations for the single n-th order ordinary difference equation.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k) \tag{27}$$

$$y(k) = \begin{bmatrix} b_n - a_n b_0 & b_{n-1} - a_{n-1} b_0 & b_{n-2} - a_{n-2} b_0 & \cdots & b_1 - a_1 b_0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} + [b_0]\ u(k)$$

This is called the *controllable canonical companion matrix* state space realization of the system described by differnce equation (24) or transfer function (25). The coefficients $a_1, \cdots, a_n$ or, equivalently, the roots $\lambda_1, \cdots, \lambda_n$, of the characteristic equation, and the input coefficients $b_0, ..., b_n$ fully specify the system.

Note the similarities and differences between the continuous time differential equation (17) and the discrete time difference equation (24), the Laplace domain transfer function (19) and the z-domain transfer function (24), and the continuous time state-space controllable canonical companion matrix realization (21) and the discrete time state-space controllable canonicla companion matrix realization (27)

Here is the observable canonical companion matrix realization of the discrete time finite difference equation, presented without the cumbersome derivation.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_{n-1}(k+1) \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ \vdots \\ b_{n-2} - a_{n-2} b_0 \\ b_{n-1} - a_{n-1} b_0 \end{bmatrix} u(k) \tag{28}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} + [b_0]\; u(k)$$

This section shows that any linear difference equation in which the order of the transfer function's numerator polynomial does not exceed the order of the denominator polynomial may be expressed as a state space model.

Following sections show how linear time invariant state space models can be analyzed and simulated.

6 Free State Response

The free state response $x(t)$ of $\dot{x}(t) = Ax(t)$ to an initial state $x(0)$ is

$$x(t) = e^{At}x(0) \tag{29}$$

where e^{At} is called the *matrix exponential.*
In the matlab language, `x(:,p)=expm(A*t(p))*xo;`

The j-th column of the matrix of free state responses $X(t) = e^{At}I_n$ is the set of responses of each states x_i, $i = 1, ..., n$ from an initial state $x_j(0) = 1$ and $x_k(0) = 0$ for all $k \neq j$.

The i-th row of the matrix of free state responses $X(t) = e^{At}I_n$ is the set of responses of the i-th state, from each initial state $x_j(0) = 1$ and $x_k(0) = 0$ for all $k \neq j$, and $j = 1, ..., n$. .

7 Free Output Response

The free output response $y(t)$ of $\dot{x}(t) = Ax(t)$ to an initial state $x(0)$ is

$$y(t) = Ce^{At}x(0) \ , \tag{30}$$

In the matlab language, `y(:,p)=C*expm(A*t(p))*xo;`

The j-th column of the matrix of free output responses $Y(t) = Ce^{At}I_n$ is the set of responses of each output from an initial condition $x_j(0) = 1$ and $x_k(0) = 0$ for all $k \neq j$.

The i-th row of the matrix of free output responses $Y(t) = Ce^{At}I_n$ is the set of responses of the i-th output, from each initial condition $x_j(0) = 1$ and $x_k(0) = 0$ for all $k \neq j$, and $j = 1, ..., n$. .

8 Unit Impulse Response Function

If the system is forced by a unit impulse $\delta(t)$ acting only on the j-th input $(u(t) = e_j\delta(t))$ the solution to $\dot{x}(t) = Ax(t) + Bu(t)$, $x(0) = 0$, for $t \geq 0$ is

$$x(t) = e^{At}Be_j \ , \tag{31}$$

and the corresponding output response is

$$y(t) = Ce^{At}Be_j \ . \tag{32}$$

The *set* of $n \times r$ unit impulse state responses, each corresponding to impulse responses from each input individually, is

$$X(t) = e^{At}B \ , \tag{33}$$

and the corresponding *set* of $m \times r$ output responses is called the system's *unit impulse response function*

$$H(t) = Ce^{At}B \ . \tag{34}$$

The i, j element of $H(t)$ is the response of output i due to a unit impulse at input j. Note that the impulse response is a special case of the free response. In other words, if there is a vector v such that $x_o = Bv$, the free response and the impulse response are equivalent. In other words, the input matrix B forms a basis for the initial condition that produces the same free response as the unit impulse response. Note that there can be initial conditions x_o which do not equal Bv because B does not necessarily span $\mathbb{R}^n$. The set of free responses can therefore be much richer than the set of impulse responses.

In the matlab language, `H(p,:,:)=C*expm(A*t(p))*B;`

9 The Dirac delta function

The unit impulse $\delta(t)$ is the symmetric unit Dirac delta function. Each Dirac delta function is zero for $t < \epsilon$ and $t > \epsilon$ and has the following properties:

$$\begin{aligned}
\int_{-\epsilon}^{\epsilon} \delta(t)\ dt &= 1 \\
\int_{-\epsilon}^{0} \delta(t)\ dt &= \frac{1}{2} \\
\int_{0}^{\epsilon} \delta(t)\ dt &= \frac{1}{2} \\
\delta(0) &= \infty \\
\int_{-\epsilon}^{\epsilon} \delta(t-\tau)\ f(\tau)\ d\tau &= f(t) \\
\int_{-\epsilon}^{0} \delta(t-\tau)\ f(\tau)\ d\tau &= \frac{1}{2} f(t) \\
\int_{0}^{\epsilon} \delta(t-\tau)\ f(\tau)\ d\tau &= \frac{1}{2} f(t)
\end{aligned}$$

10 Forced State and Output Response

The forced state response is the convolution of the inputs with the unit impulse state response function

$$x(t) = \int_0^t e^{A(t-\tau)}\ Bu(\tau)\ d\tau\ . \tag{35}$$

The output corresponding to this input is

$$\begin{aligned}
y(t) &= Cx(t) + Du(t) \\
&= C \int_0^t e^{A(t-\tau)}\ Bu(\tau)\ d\tau\ + Du(t).
\end{aligned} \tag{36}$$

The total response of a linear time invariant system from an arbitrary initial condition is the sum of the free response and the forced response.

$$y(t) = Ce^{At}x_o + C \int_0^t e^{A(t-\tau)}\ Bu(\tau)\ d\tau\ + Du(t). \tag{37}$$

An efficient method for computing $y(t)$ for a arbitrary inputs $u(t)$ is provided in the last sections of this document.

11 The Matrix Exponential

The matrix exponential is defined for $A \in \mathbb{R}^{n\times n}$ as

$$e^A \triangleq \sum_{k=0}^{\infty} A^k/k! = I + A + AA/2 + AAA/6 + AAAA/24 + \cdots \tag{38}$$

Properties of the matrix exponential:

- If $A = A^\mathsf{T}$ then $e^A > 0$.
- If $A = -A^\mathsf{T}$ then $[e^A][e^A]^\mathsf{T} = I$
- $e^{At}\, e^{Bt} = e^{(A+B)t}$
- $[e^A]^\mathsf{T} = e^{A^\mathsf{T}}$
- $[e^A]^{-1} = e^{-A}$
- $e^{T^{-1}AT} = T^{-1}e^AT$ for any square invertible matrix $T \in \mathbb{R}^{n\times n}$.

$$\begin{aligned}
e^{T^{-1}AT} &= I + T^{-1}AT + \frac{1}{2}T^{-1}AT\;T^{-1}AT + \frac{1}{6}T^{-1}AT\;T^{-1}AT\;T^{-1}AT + \cdots \\
&= T^{-1}T + T^{-1}AT + \frac{1}{2}T^{-1}AAT + \frac{1}{6}T^{-1}AAAT + \cdots \\
&= T^{-1}[I + A + AA/2 + AAA/6 + \cdots]T \\
e^{T^{-1}AT} &= T^{-1}e^AT.
\end{aligned} \tag{39}$$

- $\det(e^{At}) = e^{\mathrm{trace}(At)}$
- $e^{At}A^{-1} = A^{-1}e^{At}$
- $\mathrm{rank}(e^{At}) = n$ for any $A \in \mathbb{R}^{n\times n}$, regardless of the rank of A.
- $\frac{d}{dt}e^{At} = Ae^{At}$

$$\begin{aligned}
e^{At} &= I + At + \frac{1}{2}AAt^2 + \frac{1}{6}AAAt^3 + \frac{1}{24}AAAAt^4 + \cdots \\
\frac{d}{dt}e^{At} &= A + AA + \frac{1}{2}AAAt^2 + \frac{1}{6}AAAAt^3 + \cdots \\
&= A[I + At + \frac{1}{2}AAt^2 + \frac{1}{6}AAAt^3 + \frac{1}{24}AAAAt^4 + \cdots] \\
\frac{d}{dt}e^{At} &= Ae^{At}
\end{aligned} \tag{40}$$

- $A \int_0^t e^{A\tau} d\tau = e^{At} - I$

$$\begin{aligned} A \int_0^t e^{A\tau} d\tau &= \int_0^t \frac{d}{d\tau} e^{A\tau} d\tau \\ &= e^{A\tau}\Big|_0^t \\ &= e^{At} - e^{A0} \\ &= e^{At} - I \\ A \int_0^t e^{A\tau} d\tau &= e^{At} - I \quad (41) \\ -A \int_0^t e^{-A\tau} d\tau &= e^{-At} - I \quad (42) \end{aligned}$$

- $\int_0^t e^{A(t-\tau)} d\tau = A^{-1}(e^{At} - I)$

$$\begin{aligned} \int_0^t e^{A(t-\tau)} d\tau &= \int_0^t e^{At} e^{-A\tau} d\tau \\ &= e^{At} \int_0^t e^{-A\tau} d\tau \\ &= e^{At}(-A^{-1})\left(e^{-At} - I\right) \\ &= -e^{At}A^{-1}e^{-At} + e^{At}A^{-1} \\ &= -A^{-1}Ae^{At}A^{-1}e^{-At} + A^{-1}Ae^{At}A^{-1} \\ &= -A^{-1}e^{AAA^{-1}t}e^{-At} + A^{-1}e^{AAA^{-1}t} \\ &= -A^{-1}e^{At}e^{-At} + A^{-1}e^{At} \\ &= -A^{-1} + A^{-1}e^{At} \\ &= A^{-1}\left(e^{At} - I\right) \quad (43) \\ &= \int_0^t e^{A\tau} d\tau \quad (44) \end{aligned}$$

- for $x(0) = 0_n$, $u(t) = I_r \ \forall \ t > 0$, $\lim_{t\to\infty} e^{At} = 0_{n\times n}$, and $D = 0_{m\times r}$

$$\begin{aligned} Y(t) &= C \int_0^t e^{A(t-\tau)} d\tau \ B \\ &= CA^{-1}(e^{At} - I)B \end{aligned}$$

where $Y_{ij}(t)$ is the unit step response of output i to a unit step input on input j. The final value of outputs to unit step inputs is
$Y(\infty) = -CA^{-1}B$.
This is a result of the *final value theorem*.

With definitions of the natural frequency, $\omega_\mathrm{n} \triangleq \sqrt{k/m}$, the damping ratio, $\zeta \triangleq c/(2\sqrt{mk})$, and the damped natural frequency, $\omega_\mathrm{d} \triangleq \omega_\mathrm{n}\sqrt{|\zeta^2 - 1|}$, let

$$A = \left[\begin{array}{cc} 0 & 1 \\ -k/m & -c/m \end{array} \right] = \left[\begin{array}{cc} 0 & 1 \\ -{\omega_\mathrm{n}}^2 & -2\zeta\omega_\mathrm{n} \end{array} \right] . \qquad (45)$$

For this dynamics matrix, the matrix exponential depends on the damping ratio, ζ, as follows:

damping	*damping ratio*	e^{At}
undamped	$\zeta = 0$	$e^{At} = \left[\begin{array}{cc} \cos\omega_\mathrm{n} t & \frac{1}{\omega_\mathrm{n}} \sin\omega_\mathrm{n} t \\ -\omega_\mathrm{n}\sin\omega_\mathrm{n} t & \cos\omega_\mathrm{n} t \end{array} \right]$
under-damped	$0 < \zeta < 1$	$e^{At} = e^{-\zeta\omega_\mathrm{n} t} \left[\begin{array}{cc} \cos\omega_\mathrm{d} t + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin\omega_\mathrm{d} t & \frac{1}{\omega_\mathrm{d}} \sin\omega_\mathrm{d} t \\ -\frac{\omega_\mathrm{d}}{\sqrt{1-\zeta^2}} \sin\omega_\mathrm{d} t & \cos\omega_\mathrm{d} t - \frac{\zeta}{\sqrt{1-\zeta^2}} \sin\omega_\mathrm{d} t \end{array} \right]$
critically damped	$\zeta = 1$	$e^{At} = e^{-\omega_\mathrm{n} t} \left[\begin{array}{cc} 1 + \omega_\mathrm{n} t & t \\ -{\omega_\mathrm{n}}^2 t & 1 - \omega_\mathrm{n} t \end{array} \right]$
over-damped	$\zeta > 1$	$e^{At} = e^{-\zeta\omega_\mathrm{n} t} \left[\begin{array}{cc} \cosh\omega_\mathrm{d} t + \frac{\zeta}{\sqrt{1-\zeta^2}} \sinh\omega_\mathrm{d} t & \frac{1}{\omega_\mathrm{d}} \sinh\omega_\mathrm{d} t \\ -\frac{\omega_\mathrm{d}}{\sqrt{\zeta^2-1}} \sinh\omega_\mathrm{d} t & \cosh\omega_\mathrm{d} t - \frac{\zeta}{\sqrt{\zeta^2-1}} \sinh\omega_\mathrm{d} t \end{array} \right]$

12 Transformation of state space realizations

The input-output relationship of LTI systems by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \ , \qquad x(0) = x_o \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

to transformations in the state coordiante system. Expressing the LTI system above in terms of transformed state vector $\bar{x}(t)$, such that $x(t) = T\bar{x}(t)$ via a square full rank state transformtation matrix T,

$$\begin{aligned}T\dot{\bar{x}}(t) &= AT\bar{x}(t) + Bu(t) \ , \qquad T\bar{x}(0) = T\bar{x}_o \\ y(t) &= CT\bar{x}(t) + Du(t)\end{aligned}$$

or

$$\begin{aligned}\dot{\bar{x}}(t) &= \bar{A}\bar{x}(t) + \bar{B}u(t) \ , \qquad \bar{x}(0) = \bar{x}_o \\ y(t) &= \bar{C}\bar{x}(t) + Du(t)\end{aligned}$$

where $\bar{A} = T^{-1}AT$, $\bar{B} = T^{-1}B$, and $\bar{C} = CT$.

13 Eigenvalues and Diagonalization

Consider the dynamics matrix A of a linear time invariant, (LTI) system. If the input, $u(t)$ is zero, then $\dot{x} = Ax$. Assuming a solution of the form $x(t) = \bar{x}e^{\lambda t}$, and substituting this solution into $\dot{x} = Ax$, results in:

$$\bar{x}\lambda e^{\lambda t} = A\bar{x}e^{\lambda t}$$

or

$$\bar{x}\lambda = A\bar{x} \ , \tag{46}$$

which is a standard eigenvalue problem, in which $\bar{x}$ is an *eigenvector* and λ is the corresponding *eigenvalue*. If A is a $n \times n$ matrix, then there are n (possibly non-unique) eigenvalues $\lambda_1, \cdots, \lambda_n$ and n associated unique eigenvectors, $\bar{x}_1, \cdots, \bar{x}_n$. For the dynamics matrix given in equation (9), there are two eigenvalues.

$$\begin{aligned} \lambda_{1,2} &= -\frac{c}{2m} \pm \sqrt{\frac{c^2}{4m} - \frac{k}{m}} && (47) \\ &= -\zeta\omega_{\rm n} \pm \omega_{\rm n}\sqrt{\zeta^2 - 1} && (48) \end{aligned}$$

The dynamics matrix contains all the information required to determine the natural frequencies and damping ratios of the system.

The n eigenvectors can be assembled, column-by-column into a matrix,

$$\bar{X} = [\bar{x}_1 \ \bar{x}_2 \ \cdots \ \bar{x}_n] \ .$$

Pre-multiplying the eigen-problem by $\bar{X}^{-1}$,

$$\bar{X}^{-1}A\bar{X} = \bar{X}^{-1}\bar{X}\Lambda = \mathsf{diag}(\lambda_i) = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \tag{49}$$

This is called a *diagonalization* of the dynamics matrix A.

Now, consider the linear transformation of coordinates, $x(t) = \bar{X}q(t)$, $q(t) = \bar{X}^{-1}x(t)$. Substituting this change of coordinates into equation (3),

$$\bar{X}\dot{q}(t) = A\bar{X}q(t) + Bu(t),$$

and pre-multiplying by $\bar{X}^{-1}$

$$\dot{q} = \bar{X}^{-1} A \bar{X} q + \bar{X}^{-1} B u,$$

or

$$\begin{aligned}
\dot{q} &= \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} q + \bar{X}^{-1} B u \ , \qquad q(0) = \bar{X}^{-1} x_o \\
\dot{q} &= \bar{A} q + \bar{B} u \ , \qquad q(0) = \bar{X}^{-1} x_o && (50) \\
y &= C \bar{X} q + D u \\
y &= \bar{C} q + D u && (51)
\end{aligned}$$

The n differential equations $\dot{q}_i = \lambda_i q_i + \bar{B}_i u$ are *uncoupled*. The state $q_i(t)$ is independent of all the other states $q_j(t), j \neq i$. Note that if Λ is complex, so are $\bar{X}$, $\bar{A}$, $\bar{B}$, and $\bar{C}$.

Consider one of the un-coupled equations from equation (50), for the unforced case $u = 0$

$$\dot{q}_i(t) = \lambda_i q_i(t) \ , \qquad q_i(0) = 1 \ .$$

This equation has a solution

$$q_i(t) = e^{\lambda_i t} \ ,$$

where λ_i is, in general, a complex value,

$$\lambda_i = \sigma_i \pm \mathrm{i}\omega_i \ ,$$

and $q_i + q_i^*$ is real-valued.

$$\begin{aligned}
q_i(t) + q_i^*(t) &= \frac{1}{2} e^{\lambda_i t} + \frac{1}{2} e^{\lambda_i^* t} \\
&= \frac{1}{2} e^{\sigma_i t} (\cos \omega_i t + \mathrm{i} \sin \omega_i t) + \frac{1}{2} e^{\sigma_i t} (\cos \omega_i t - \mathrm{i} \sin \omega_i t) \\
&= e^{\sigma_i t} \cos \omega_i t \ ,
\end{aligned}$$

14 Jordan Forms

If a square matrix $A \in \mathbb{R}^{n\times n}$ has *distinct* eigenvalues, then it can be reduced to a diagonal matrix through a similarity transformation

$$\bar{X}^{-1} \, A \, \bar{X} = \Lambda = \left[\begin{array}{ccc} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{array} \right] \tag{52}$$

where $\bar{X} \in \mathbb{R}^{n\times n}$ is the matrix of *linearly independent* eigenvectors of A.

If A has *repeated* eigenvalues, the it can be reduced to a diagonal matrix only if all n eigenvectors are linearly independent.

If A has *repeated* eigenvalues and two or more of the eigenvectors associated with the repeated eigenvalues are not linearly independent, then it is not similar to a diagonal matrix. It is, however, similar to a simpler matrix called the *Jordan form* of A,

$$\bar{X}^{-1} \, A \, \bar{X} = \left[\begin{array}{ccc} J_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_n \end{array} \right] \tag{53}$$

where the square sub-matrices in

$$J_i = \left[\begin{array}{cccccc} \lambda_i & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \lambda_i & 1 \\ 0 & \cdots & \cdots & 0 & 0 & \lambda_i \end{array} \right] \tag{54}$$

are called *Jordan blocks.* In a Jordan block, repeated eigenvalues are on the diagonal and 1's are just above the diagonal.

Consider an eigenvalue λ_i of matrix $A \in \mathbb{R}^{n\times n}$ with multiplicity k. There are k eigenvectors, $\bar{x}_i$, associated with the eigenvalue λ_i. If *all* pairs of these

k eigenvectors are linearly dependent, (there exist real values α_i such that $\bar{x}_1 = \alpha_i \bar{x}_i \quad i = 2, \ldots, k$), then this set of linearly dependent eigenvectors span a one-dimensional subspace, and a single Jordan block of dimension $k \times k$ will be associated with the eigenvalue λ_i. On the other hand, if the eigenvalue λ_i has b_i linearly independent eigenvectors, then there will be b_i separate Jordan blocks associated with λ_i.

The similarity transformation into Jordan form $\bar{X}^{-1} A \bar{X} = J$ may be written $A \bar{X} = \bar{X} J$, in which the columns of $\bar{X}$ are $[\bar{x}_1 \ \bar{x}_2 \ \cdots \bar{x}_n]$. So

$$A[\bar{x}_1 \ \bar{x}_2 \ \cdots \ \bar{x}_n] = [\bar{x}_1 \ \bar{x}_2 \ \cdots \ \bar{x}_n]J \tag{55}$$

Writing the columns of $\bar{X}$ associated with a single Jordan block of λ_i,

$$\begin{aligned}
A[u_1 \ u_2 \ u_3 \ \cdots \ u_k] &= [u_1 \ u_2 \ u_3 \ \cdots \ u_k] \begin{bmatrix} \lambda_i & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_i & 1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_i & 1 & \cdots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \lambda_i & 1 \\ 0 & \cdots & \cdots & 0 & 0 & \lambda_i \end{bmatrix}_{k \times k} \\
[Au_1 \ Au_2 \ Au_3 \ \cdots \ Au_k] &= [u_1\lambda_i \quad u_1 + u_2\lambda_i \quad u_2 + u_3\lambda_i \quad \cdots \quad u_{k-1} + u_k\lambda_i],
\end{aligned}$$

and associating columns of the left and right hand side of this equation,

$$\begin{aligned}
Au_1 = u_1\lambda_i &\Leftrightarrow [A - \lambda_i I]u_1 = 0 \\
Au_2 = u_1 + u_2\lambda_i &\Leftrightarrow [A - \lambda_i I]u_2 = u_1 \\
Au_3 = u_2 + u_3\lambda_i &\Leftrightarrow [A - \lambda_i I]u_3 = u_2 \Leftrightarrow [A - \lambda_i I]^2 u_3 = u_1 \\
\vdots \qquad &\quad \vdots \\
Au_k = u_{k-1} + u_k\lambda_i &\Leftrightarrow [A - \lambda_i I]u_k = u_{k-1} \Leftrightarrow [A - \lambda_i I]^{k-1} u_k = u_1
\end{aligned}$$

The size of the Jordan block is $k \times k$ where u_k is not linearly independent of u_{k+1}. An equivalent criterion is

$$\text{rank}[A - \lambda_i I]^k = \text{rank}[A - \lambda_i I]^{k+1}. \tag{56}$$

So, to find the vectors of the transformation matrix $\bar{X}$ in a particular Jordan block of λ_i, start first by finding k such that the above equation is satisfied.

The next step is to find a vector in the null space of $[A-\lambda_i I]^k$. start by setting u_1 equal to one of the eigenvectors of λ_i. Then iterate on

$$u_n = [A-\lambda_i I]^{-1} u_{n-1} \tag{57}$$

until u_n and u_{n-1} are not linearly independent. If there is more than one Jordan block associated with the repeated eigenvalue λ_i, the transformation vectors associated with the remaining Jordan blocks may be found by restarting the iterations with the other eigenvectors associated with λ_i.

If $A \in \mathbb{R}^{n\times n}$ and k is the multiplicity of the eigenvalue λ_i, then the size of the largest Jordan block of eigenvalue λ_i is k such that

$$\text{rank}(A-\lambda_i I)^k = \text{rank}(A-\lambda_i I)^{k+1}. \tag{58}$$

For a similar derivation of this see pages 42–49 of C-T Chen, *Introduction to Linear Systems Theory,* Hold, Rinehart and Winston, 1970.

If $\hat{A} \in \mathbb{R}^{n\times n}$ has the Jordan block form

$$\begin{bmatrix} \lambda_1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 & 1 & \cdots & 0 \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \lambda_1 & 1 \\ 0 & \cdots & \cdots & 0 & 0 & \lambda_1 \end{bmatrix}_{n\times n} \tag{59}$$

then

$$e^{\hat{A}t} = \begin{bmatrix} e^{\lambda_1 t} & t\ e^{\lambda_1 t} & t^2\ e^{\lambda_1 t}/2! & t^3\ e^{\lambda_1 t}/3! & \cdots & t^{n-1}\ e^{\lambda_1 t}/(n-1)! \\ 0 & e^{\lambda_1 t} & t\ e^{\lambda_1 t} & t^2\ e^{\lambda_1 t}/2! & \cdots & t^{n-2}\ e^{\lambda_1 t}/(n-2)! \\ 0 & 0 & e^{\lambda_1 t} & t\ e^{\lambda_1 t} & \cdots & t^{n-3}\ e^{\lambda_1 t}/(n-3)! \\ \vdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & e^{\lambda_1 t} & t\ e^{\lambda_1 t} \\ 0 & \cdots & \cdots & 0 & 0 & e^{\lambda_1 t} \end{bmatrix}_{n\times n} \tag{60}$$

15 Transfer Function and Frequency Response Function

Taking the Laplace transform of equation (3), (and considering the particular part of the solution, (i.e., ignoring the effects of initial conditions) gives

$$s\ x(s) = Ax(s) + Bu(s)\ , \tag{61}$$

$$y(s) = Cx(s) + Du(s)\ , \tag{62}$$

which can be written as $y(s)$ in terms of $u(s)$ as follows,

$$y(s) = H(s)\ u(s) = [\ C[sI - A]^{-1}B + D\ ]\ u(s) \tag{63}$$

This *transfer function* relates the set of r inputs u to the m outputs y in the Laplace domain; $H(s) \in \mathbb{C}^{m \times r}$.

Equation (63) may be used to determine the complex-valued *frequency response function* of any dynamic system, by evaluating the transfer function along the line $s = \mathrm{i}\omega$.

$$y(\omega) = H(\omega)\ u(\omega) = [C[\mathrm{i}\omega I - A]^{-1}B + D]\ u(\omega) \tag{64}$$

Assuming that the inputs $u(t)$ are sinusoidal with frequency ω and unit amplitude, the magnitude of the frequency response function,

$$|H(\omega)| = \sqrt{\mathsf{Re}[H(\omega)]^2 + \mathsf{Im}[H(\omega)]^2}$$

gives the amplitude of the responses $y(t)$. The phase of $H(\omega)$,

$$\angle H(\omega) = \arctan\left(\frac{\mathsf{Im}[H(\omega)]}{\mathsf{Re}[H(\omega)]}\right)$$

gives the phase angle between the input $u(t)$ and the output $y(t)$. So if $u(t) = \cos(\omega t)$, then $y(t) = |H(\omega)| \cos(\omega t + \angle H(\omega))$. A graph of $|H(\omega)|$ and $\angle H(\omega)$ is called a *Bode plot*. In the matlab language, `[mag,pha] = bode(A,B,C,D);` generates a Bode plot for a system defined by matrices A, B, C, and D.

Transfer functions (and frequency response functions) are invariant to coordinate transformation. Any transformed equation of state, $\bar{A} = T^{-1}AT$,

$\bar{B} = T^{-1}B$, $\bar{C} = CT$ (e.g., equations (50) and (51)), resulting in

$$y(s) = H(s)\ u(s) = [\bar{C}[sI - \bar{A}]^{-1}\bar{B} + D]\ u(s).$$

If $T = \bar{X}$, the eigenvector matrix of A (see equation (49)), then $\bar{A} = \Lambda = \mathsf{diag}(\lambda_i)$. Continuing with a realization in modal coordinates, in which $\bar{B} = \bar{X}^{-1}B$ and $\bar{C} = C\bar{X}$,

$$H(s) = \begin{bmatrix} - & \bar{c}_1 & - \\ & \vdots & \\ - & \bar{c}_m & - \end{bmatrix} \begin{bmatrix} \frac{1}{s-\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{s-\lambda_n} \end{bmatrix} \begin{bmatrix} | & & | \\ \bar{b}_1 & \cdots & \bar{b}_r \\ | & & | \end{bmatrix} + D.$$

in which $\bar{c}_i$ is the i^{th} row of $\bar{C}$ and $\bar{b}_i$ is the j^{th} column of $\bar{B}$. Since Λ is diagonal, we can express the i, j element of the transfer function matrix $H(s)$ as

$$H_{ij}(s) = \sum_{k=1}^{n} \frac{\bar{C}_{ik}\ \bar{B}_{kj}}{s - \lambda_k} + D_{ij} \tag{65}$$

Putting equation (66) over a common denominator, $H_{ij}(s)$ becomes

$$H_{ij}(s) \ = \ g_{ij} \frac{\prod_{k=1}^{p} (s - z_k^{(ij)})}{\prod_{k=1}^{n} (s - \lambda_k)} \tag{66}$$

where the leading coefficient g_{ij} and the zeros of the numerator $z_k^{(ij)}$ depend algebraically upon $\bar{c}_i$, $\bar{b}_j$ and $(\lambda_1, ..., \lambda_n)$. Now expanding the numerator and denominator products, $H_{ij}(s)$ may be written in *Prony series* form,

$$H_{ij}(s) \equiv \frac{\bar{y}_i(s)}{\bar{u}_j(s)} = \frac{b_0^{(ij)} + b_1^{(ij)}s + b_2^{(ij)}s^2 + \cdots + b_{p-1}^{(ij)}s^{n-1} + b_p^{(ij)}s^n}{a_0^{(ij)} + a_1^{(ij)}s + a_2^{(ij)}s^2 + \cdots + a_{n-1}^{(ij)}s^{n-1} + s^n} \tag{67}$$

where the numerator coefficients $b_k^{(ij)}$ depend upon g_{ij} and the zeros $z_k^{(ij)}$, whereas the denominator coefficients a_k depend only upon the eigenvalues $(\lambda_1, ..., \lambda_n)$. For any LTI system, every element of a transfer function matrix has the same denominator polynomial. Note that $p \leq n$, and that $D_{ij} \neq 0 \Leftrightarrow p = n$. If and only if $D = 0$ $(p < n)$, the LTI system is called *strictly proper.*

16 Singular Value Spectra

In systems with multiple inputs and multiple outputs (MIMO), the strength of the output depends upon the relative amplitudes and phases of the inputs. It is possible that the effect of two non-zero inputs properly scaled in amplitude and phase, could have a relatively small effect on the responses, while a different combination if inputs could have a very strong affect on the system response. Further, for harmonic inputs (and outputs), the scaling and phasing of the inputs to achieve a strong or weak response is dependent upon the frequency.

The singular value decomposition of the frequency response function matrix

$$\begin{aligned} y(\omega) &= H(\omega)_{(m\times r)}\ u(\omega) \\ &= U_H(\omega)\ \Sigma_H(\omega)\ V_H^{\mathsf{T}}(\omega)\ u(\omega) \\ &= \sum_k \sigma_{Hk}(\omega)[u_{Hk}(\omega)v_{Hk}^{\mathsf{T}}(\omega)]\ u(\omega) \\ &= \sum_k \sigma_{Hk}(\omega)u_{Hk}(\omega)\ (v_{Hk}^{\mathsf{T}}(\omega)\ u(\omega)) \end{aligned} \tag{68}$$

provides the means to assess how inputs can be scaled for maximal or minimal effect. Here, columns of $U_H(\omega)$ and $V_H(\omega)$ are the left and right singular vectors of $H(\omega)$. For complex-valued frequency responses, singular vectors are complex-valued. (The singular values are always real ($\sigma_{H1} \geq \sigma_{H2} \geq ... \geq 0$)). For inputs $u(\omega)$ proportional to $v_{H1}(\omega)$, $||y(\omega)||_2$ is maximized and is proportional to $u_{H1}(\omega)$. If $r < m$ and $\sigma_{Hr} > 0$ then inputs proportional to $V_{Hr}(\omega)$ have the weakest coupling to the outputs. On the other hand, if $r > m$ there is some linear combination of inputs that (at a particular frequency ω) will result in no output at all. Right singular vectors spanning this kernel of $H(\omega)$, $[v_{H(m+1)}(\omega)\ ,\ ...\ ,\ v_{Hr}(\omega)]$, is an orthogonal basis for these inputs

Plots of $\sigma_{H1}(\omega)$ and $\sigma_{Hr}(\omega)$ indicate the frequency-dependence of the largest amplification and the smallest amplification for any linear combination (and phasing) of inputs, in the context of steady-state harmonic response.

17 Zeros and Poles of MIMO LTI Systems

In the Laplace domain, the dynamics equation is $sx(s) = Ax(s) + Bu(s)$, or

$$\left[\; sI - A \; , \;\; -B \; \right]_{n\times(n+r)} \left[\begin{array}{c} x(s) \\ u(s) \end{array} \right] = 0 \; .$$

If there is a value of s such that

$$\text{rank}([sI - A \; , \; B]) < n \; ,$$

then there is a non-zero $u(s)$ for which $x(s)$ can not be uniquely determined ($x(s)$ could be zero in one or more components).

Similarly, the output response in the Laplace domain is $sx(s) = Ax(s)$, $y(s) = Cx(s)$, or

$$\left[\begin{array}{c} sI - A \\ C \end{array} \right]_{(n+m)\times n} x(s) = \left[\begin{array}{c} 0 \\ y(s) \end{array} \right] \; .$$

If there is a value of s such that

$$\text{rank} \left(\left[\begin{array}{c} sI - A \\ C \end{array} \right] \right) < n$$

then the non-trivial null space of this matrix, $x(s)$, corresponds to $y(s) = 0$. In other words, there is a subspace of the state-space that does not couple to the output.

These rank conditions for controllability and observability are called the Popov-Belevitch-Hautus (PBH) tests. Note that for any $s \neq \lambda_i$ (the eigenvalues of A), $\text{rank}(sI - A) = n$, and that $\text{rank}(\lambda_i I - A) < n$. If the columns of B do not span the $\mathcal{N}(\lambda_i I - A)$, then $u(s)$ does not couple to the i-th mode of the system, and λ_i is an *input-decoupling zero*. If a system has one or more input decoupling zeros, then there are inputs that can not affect a subspace of the state space, and the system is called *uncontrollable*. Similarly, if the rows of C do not span $\mathcal{N}(\lambda_i I - A^{\mathsf{T}})$, then $y(s)$ does not couple to the i-th mode of the system, and λ_i is an *output-decoupling zero*. If a system has one or more output decoupling zeros, then there are states that do not affect the output, and the system is called *unobservable.*

The matrix-valued MIMO transfer function may be represented as a system of two sets of linear equations in the Laplace domain

$$\begin{aligned} x(s) &= (sI - A)^{-1} B u(s) \\ y(s) &= C x(s) + D u(s) \ , \end{aligned} \tag{69}$$

which is equivalent to

$$\begin{bmatrix} sI - A & -B \\ C & D \end{bmatrix} \begin{bmatrix} x(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} 0 \\ y(s) \end{bmatrix} . \tag{70}$$

This is called the Rosenbrock System Matrix (RSM) formulation. For a system with a zero output,

$$\left[\begin{bmatrix} -A & -B \\ C & D \end{bmatrix} + s \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix} \right] \begin{bmatrix} x(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{71}$$

which is a generalized eigenvalue problem when D is square. Note that in this eigenvalue problem, the matrix multiplying the eigenvalue s is not invertible, and requires a numerical method such as the QZ decomposition.[1] Eigenvectors corresponding to finite values of s satisfying this generalized eigenvalue problem, define the magnitudes and phases of inputs $u(s)$ (and of the corresponding states $x(s)$), such that the output $y(s)$ is zero. In other words, at values of s for which the rank of the Rosenbrock System Matrix is less than $(n + \min(\text{rank}B, \text{rank}C))$, there exists a set of non-zero inputs such that the output is zero.

Values of s satisfying the generalized eigenvalue problem (71) are called *invariant zeros.*[2]

Eigenvalues of A that are not also zeros are called *poles.*

[1] C.B. Moler and G.W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM J. Numer. Anal.*, 10(2) (1973), 241-256

[2] H.H. Rosenbrock, "The zeros of a system," *Int'l J. Control*, 18(2) (1973): 297-299.

18 Liapunov Stability

Consider autonomous dynamic systems which may be linear $\dot{x} = Ax$ or nonlinear $\dot{x} = f(x)$ evolving from an initial condition $x(0) = x_o$. The equilibrium solution $x_e(t) = 0$ satisfies the linear system dynamics $\dot{x} = Ax$, and satisfies the nonlinear system dynamics $\dot{x} = f(x)$ if (and only if) $f(0) = 0$. Going forward we will suppose that $x(t) = 0$ is a solution.

18.1 Classifications of the stability of equilibrium solutions $x_e(t) \equiv 0$

The equilibrium solution $x_e(t) \equiv 0$ is:

- *Liapunov Stable* (LS) if and only if

$$\forall \;\; \epsilon > 0 \; , \quad \exists \; \delta > 0 \quad \text{such that} \quad \forall \; ||x_o|| < \delta \Rightarrow ||x(t)|| < \epsilon \; \forall \; t \geq 0$$

- *Globally Semi Stable* (GSS) if and only if

$$\lim_{t \to \infty} x(t) \;\; \text{exists} \;\; \forall \;\; x_o$$

- *Locally Semi Stable* (LSS) if and only if

$$\exists \; \epsilon > 0 \;\; \text{such that} \;\; \forall \; ||x_o|| < \epsilon \;\; \Rightarrow \;\; \lim_{t \to \infty} x(t) \; \text{exists}$$

- *Asymptotically Stable* (AS) if and only if

$$\exists \; \epsilon > 0 \; \text{such that} \;\; \forall \; ||x_o|| < \epsilon \;\; \Rightarrow \;\; \lim_{t \to \infty} x(t) \to 0$$

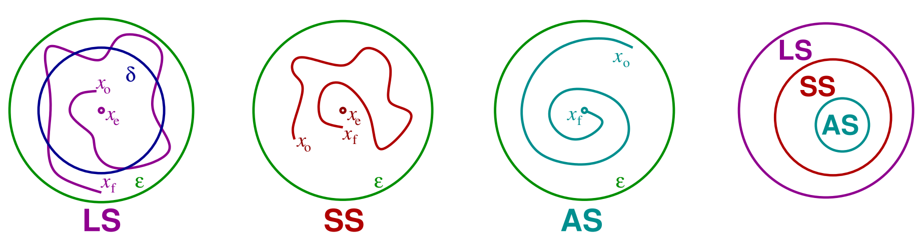

Figure 1. Classifications of the stability of equilibrium solutions $x_e(t) \equiv 0$, (AS) implies (SS) implies (LS).

18.2 Liapunov functions of solutions $x(t)$

Define a scalar-valued function of the state $V(x(t))$, $V : \mathbb{R}^n \to \mathbb{R}$, (where $\dot{x} = f(x)$). Then, by the chain rule

$$\dot{V} = \frac{d}{dt} V(x(t)) = V'(x)\dot{x}(t) = V'(x(t)f(x(t))$$

Now, let $V(x) > 0 \ \forall \ x \neq 0$, and $V(0) = 0$.

- If
$$\dot{V}(x) \leq 0 \ \forall \ x$$
then the system $\dot{x} = f(x)$ is *Liapunov Stable* (LS).

- If
$$\dot{V}(x) \leq 0 \ \forall \ x \text{ and } f(x) = 0 \ \forall x \text{ such that} \dot{V}(x) = 0$$
then the system $\dot{x} = f(x)$ is *Semi Stable* (SS).
Note: In this case $\dot{V}$ may be zero even if $V \neq 0$.

- If
$$\dot{V}(x) < 0 \ \forall \ x \text{ and } V(x) \to \infty \text{ as } ||x|| \to \infty$$
then the system $\dot{x} = f(x)$ is *Asymptotically Stable* (AS).
Note: In this case $\dot{V}$ is always negative.

18.3 Stability of Linear Systems

Consider $\dot{x} = Ax$ with $x(0) = x_o$. The equilibrium solution $x_{\mathrm{e}}(t) \equiv 0$ is:

- *Liapunov Stable* (LS) if and only if
$$\exists \ \epsilon \text{ such that} ||e^{At}|| < \epsilon \quad \forall \ t \geq 0$$

- *Semi Stable* (SS) if and only if
$$\lim_{t\to\infty} e^{At} \quad \text{exists}$$

- *Asymptotically Stable* (AS) if and only if
$$\lim_{t\to\infty} e^{At} \to 0 \text{ as } t \to \infty$$

18.4 Examples

- (not LS) : rigid body motion

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \ , \ e^{At} = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \ , \quad \text{so } ||e^{At}|| \to \infty \text{ as } t \to \infty$$

- (LS) : undamped

$$A = \begin{bmatrix} 0 & 1 \\ -{\omega_{\rm n}}^2 & 0 \end{bmatrix} \ , \ ||e^{At}||_{\rm F} \ \ \text{exists} \quad \left(\sqrt{2 + {\omega_{\rm n}}^2 + {\omega_{\rm n}}^{-2}}\right)$$

- (AS) : damped

$$A = \begin{bmatrix} 0 & 1 \\ -{\omega_{\rm n}}^2 & -2\zeta\omega_{\rm n} \end{bmatrix} \ , \ e^{At} \ \to \ 0$$

For a general two state system, $\dot{x} = Ax$,

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The dynamics A are:

- *Liapunov Stable* (LS) if and only if trace $A \le 0$, det $A \ge 0$, and rank $A =$ rank A^2
- *Semi Stable* (SS) if and only if (trace $A < 0$ and det $A \ge 0$) or $A = 0$
- *Asymptotically Stable* (AS) if and only if trace $A < 0$, det $A > 0$.

The stability classifications of general matrix second order systems have been systematically addressed. [3] A matrix second order system is

$$M\ddot{r}(t) + C\dot{r}(t) + Kr(t) = 0$$

where $r \in \mathbb{R}^n$, $M > 0$, $C \ge 0$, and $K \ge 0$,

[3] Bernstein, D.S. and Bhat, S.P., Liapunov Stability, Semi Stability, and Asymptotic Stability of Matrix Second Order Systems," *ASME Journal of Mechanical Design,* 117 (1995): 145-153.

- $K > 0 \;\Rightarrow\;$ *Liapunov Stable* (LS)
- $C > 0 \;\Rightarrow\;$ *Semi Stable* (SS)
- $K > 0$ and $C > 0 \;\Rightarrow\;$ *Asymptotically Stable* (AS)

Furthermore, the system is:

- *Liapunov Stable* (LS) if and only if: $C + K > 0$
- *Semi Stable* (SS) if and only if:
  rank $[C, \;\; KM^{-1}C, \;\; (KM^{-1})^2C, \; \cdots \; (KM^{-1})^{n-1}C] = n$
- *Asymptotically Stable* (AS) if and only if: $K > 0$ and the system is (SS).

18.5 Asymptotic Stability of LTI Systems

The dynamics matrix fully specifies the stability properties of LTI systems. A Liapunov function of the system defined by $V(x(t)) = x(t)^{\mathsf{T}} P x(t)$ where $P > 0$, is an "energy-like function." For example, for a spring-mass system with states corresponding to positions $p(t)$, and velocities $v(t)$,

$$x(t) = \left[\begin{array}{c} p(t) \\ v(t) \end{array} \right]$$

A Liapunov function

$$V(x(t)) = \frac{1}{2} \left[\begin{array}{c} p(t) \\ v(t) \end{array} \right]^{\mathsf{T}} \left[\begin{array}{cc} K & 0 \\ 0 & M \end{array} \right] \left[\begin{array}{c} p(t) \\ v(t) \end{array} \right] = \frac{1}{2} p(t)^{\mathsf{T}} K p(t) + \frac{1}{2} v(t)^{\mathsf{T}} M v(t)$$

represents the sum of the potential energy and kinetic energy. If the stiffness matrix K and the mass matrix M are both positive definite then P is also positive definite and $V(t) > 0$ for any non-zero displacements $p(t)$ and velocities $v(t)$. The rate of change of this Liapunov function is

$$\dot{V}(x(t)) = x(t)^{\mathsf{T}} P \dot{x}(t) + \dot{x}(t)^{\mathsf{T}} P x(t).$$

Substituting the free response dynamics, $\dot{x}(t) = Ax(t)$, into $\dot{V}$, results in

$$\dot{V}(x(t)) = x(t)^{\mathsf{T}} A^{\mathsf{T}} P x(t) + x(t)^{\mathsf{T}} P A x(t) = x(t)^{\mathsf{T}} [A^{\mathsf{T}} P + PA] x(t) \ .$$

So if $A^{\mathsf{T}}P + PA$ is negative definite, $\dot{V}(x(t))$ decreases monotonically, for any non-zero value of the state vector $x(t)$. The condition $0 > A^{\mathsf{T}}P + PA$ is equivalent to $0 = A^{\mathsf{T}}P + PA + R$ for a matrix $R > 0$.

The equation

$$A^{\mathsf{T}}P + PA + R = 0$$

is called a *Liapunov equation.*

The following statements are equivalent:

- A is asymptotically stable
- all eigenvalues of A have real parts that are negative
- $\exists\ R > 0$ s.t. $P > 0$ satisfies the Liapunov equation $A^{\mathsf{T}}P + PA + R = 0$.
- $\exists\ R > 0$ s.t. the integral

  $$P = \int_0^\infty e^{A^{\mathsf{T}}t} R e^{At} \ dt$$

  converges and satisfies the Liapunov equation

  $$A^{\mathsf{T}}P + PA + R = 0 \ .$$

  Proof: Substituting the integral above into the Liapunov equation, and with the presumption that A is asymptotically stable,

$$\begin{aligned}
\int_0^\infty A^{\mathsf{T}} e^{A^{\mathsf{T}}t} \ R \ e^{At} \ dt + \int_0^\infty e^{A^{\mathsf{T}}t} \ R \ e^{At} A \ dt + R &= 0 \\
\int_0^\infty \left[A^{\mathsf{T}} e^{A^{\mathsf{T}}t} \ R \ e^{At} + e^{A^{\mathsf{T}}t} \ R \ e^{At} A \right] dt + R &= 0 \\
\int_0^\infty \frac{d}{dt} \left[e^{A^{\mathsf{T}}t} \ R \ e^{At} \right] \ dt + R &= 0 \\
\left[e^{A^{\mathsf{T}}t} \ R \ e^{At} \right]_0^\infty \ + R &= 0 \\
0 \cdot R \cdot 0 - I \cdot R \cdot I + R &= 0
\end{aligned}$$

19 Observability and Controllability

If the solution P to the *left Liapunov equation*

$$0 = A^{\mathsf{T}} P + PA + C^{\mathsf{T}} C \tag{72}$$

is positive definite, then the system defined by A and C is called *observable*, meaning that the initial state can be inferred from the time series of free responses, $y(t) = Cx(t)$

The matrix of free output responses from independent initial conditions on every state, $x(0) = I_n$ is $Y(t) = Ce^{At}$. This matrix of independent free responses has m rows and n columns. The covariance of $Y^{\mathsf{T}}(t)$ is called observability gramian and solves the left Liapunov equation, above.

$$P = \int_0^\infty Y^{\mathsf{T}}(t) Y(t)\ dt = \int_0^\infty (e^{A^{\mathsf{T}}t} C^{\mathsf{T}})(Ce^{At})\ dt \tag{73}$$

If the solution Q to the *right Liapunov equation*

$$0 = AQ + QA^{\mathsf{T}} + BB^{\mathsf{T}} \tag{74}$$

is positive definite, then the system defined by A and B is called *controllable*, meaning that the controls u acting on the system $\dot{x} = Ax + Bu$ can return the system to $x = 0$ from any initial state $x(0)$ in finite time.

The matrix of state response sequence from independent impulses on each input, $u(t) = I_r \delta(t)$ is $X(t) = e^{At} B$. This matrix of independent state impulse responses has n rows and r columns. The covariance of $X(t)$ is called the *controllability gramian.* and solves the right Liapunov equation, above.

$$Q = \int_0^\infty X(t) X^{\mathsf{T}}(t)\ dt = \int_0^\infty (e^{At} B)(B^{\mathsf{T}} e^{A^{\mathsf{T}}t})\ dt \tag{75}$$

These Liapunov equations and gramians are useful in determining the norm of LTI systems.

20 H_2 norms of Continuous-Time LTI systems

Consider a stable ($\mathsf{Re}(\lambda(A)) < 0$), strictly proper ($D = 0$), MIMO LTI system, equivalently described by a state space realization

$$\dot{x}(t) = Ax(t) + Bu(t) \ , \qquad y(t) = Cx(t) \ ,$$

and its unit impulse response function matrix, $H(t) = Ce^{At}B$, or its frequency response function matrix, $H(\omega) = C(i\omega I - A)^{-1}B$. The H_2 norm defines a scalar measure the of dynamic amplification of the dynamic system[4]. There are three ways to view, motivate, define, or interpret the H_2 norm.

20.1 The Frobeneus Norm

The H_2 norm of a stable, strictly proper MIMO system is defined in terms of the Frobeneus norm. The Frobeneus norm of a matrix is the square-root of the sum of the squares of all the terms in the matrix.

$$\begin{aligned} ||A||_{\mathrm{F}} &= \left[\sum_{i,j}^{n,m} A_{ij}^2 \right]^{1/2} \\ &= [A_{1,1}^2 + A_{1,2}^2 + \cdots + A_{1,n}^2 + \\ &\quad A_{2,1}^2 + A_{2,2}^2 + \cdots + A_{2,n}^2 + \\ &\quad \cdots \\ &\quad A_{m,1}^2 + A_{m,2}^2 + \cdots + A_{m,n}^2]^{1/2} \end{aligned}$$

If a matrix A is real ($A \in \mathbb{R}^{n\times m}$) then the Frobeneus norm of A is

$$||A||_{\mathrm{F}} = \left[\mathrm{tr}\ AA^{\mathsf{T}}\right]^{1/2}$$

If a matrix A is complex ($A \in \mathbb{C}^{n\times m}$) then the Frobeneus norm of A is

$$||A||_{\mathrm{F}} = [\mathrm{tr}\ AA^*]^{1/2}$$

The Frobeneus norm of A is the same as the Frobeneus norm of A^{T}.

$$||A||_{\mathrm{F}} = ||A^{\mathsf{T}}||_{\mathrm{F}} = [\mathrm{tr}\ AA^{\mathsf{T}}]^{1/2} = [\mathrm{tr}\ A^{\mathsf{T}}A]^{1/2}$$

[4] The objective function for linear quadratic control synthesis is the H_2 norm of the closed-loop system.

20.2 The H_2 norm of a system in terms of unit impulse response

The H_2 norm of an LTI system is defined in terms of the Frobeneus norm of its matrix of unit impulse response functions $H(t)$, as the sum of the areas under all the unit impulse response functions $H_{ij}(t)$

$$
\begin{aligned}
||H||_2^2 &= \int_0^\infty ||H(t)||_{\mathrm{F}}^2 \; dt &(76)\\
&= \int_0^\infty \mathrm{tr}\; [H^\mathsf{T}(t)H(t)] \; dt \\
&= \mathrm{tr}\; \int_0^\infty (B^\mathsf{T} e^{A^\mathsf{T} t} C^\mathsf{T})(Ce^{At}B) \; dt \\
&= \mathrm{tr}\; B^\mathsf{T} \int_0^\infty (e^{A^\mathsf{T} t} C^\mathsf{T})(Ce^{At}) \; dt \; B \\
&= \mathrm{tr}\; B^\mathsf{T} P B &(77)
\end{aligned}
$$

where P is the observability gramian,

$$P = \int_0^\infty (e^{A^\mathsf{T} t} C^\mathsf{T})(Ce^{At}) \; dt$$

20.3 The H_2 norm of a system in terms of frequency response

Another interpretation of the H_2 norm is in the frequency domain. This interpretation is an expression of Parseval's Theorem,

$$\int_0^\infty ||H(t)||_{\mathrm{F}}^2 \; dt = \frac{1}{2\pi} \int_{-\infty}^\infty ||H(\mathrm{i}\omega)||_{\mathrm{F}}^2 \; d\omega \tag{78}$$

To prove that this identity is true, we first need to know four facts. First we need to know that if A is asymptotically stable, then

$$\int_0^\infty e^{At} \; dt = -A^{-1}.$$

Proof:

$$\int_0^\infty e^{At} \; dt = A^{-1} \int_0^\infty Ae^{At} \; dt = A^{-1} \int_0^\infty \left(\frac{d}{dt} e^{At} \right) \; dt = A^{-1} \; e^{At} \Big]_0^\infty = A^{-1} \left(\lim_{t\to\infty} \left(e^{At} \right) - I \right)$$

Second we need to know that the Laplace transform of e^{At} is $(sI - A)^{-1}$

$$\mathcal{L}\{e^{At}\} = \int_0^\infty e^{At} e^{-st} \; dt = \int_0^\infty e^{(A-sI)t} \; dt = (sI - A)^{-1}$$

Third we need to know that the Laplace transform of $H(t)$ is $C(sI-A)^{-1}B$

$$\mathcal{L}\{H(t)\} = \mathcal{L}\{Ce^{At}B\} = C\mathcal{L}\{e^{At}\}B = C(sI-A)^{-1}B$$

Fourth, we need to know that if Q satisfies $0 = AQ + QA^{\mathsf{T}} + BB^{\mathsf{T}}$, then Q is given by

$$Q = \frac{1}{2\pi}\int_{-\infty}^{\infty}(\mathrm{i}\omega I - A)^{-1}BB^{\mathsf{T}}(\mathrm{i}\omega I - A)^{-*}\, d\omega$$

for the special case in which A has been diagonalized, and $BB^{\mathsf{T}} = I_n$.

Proof:

$$\begin{aligned}
Q &= \frac{1}{2\pi}\int_{-\infty}^{\infty}(\mathrm{i}\omega I - A)^{-1}BB^{\mathsf{T}}(\mathrm{i}\omega I - A)^{-*}\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}[(-\mathrm{i}\omega I - A)(\mathrm{i}\omega I - A)]^{-1}\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}[\omega^2 I + A^2]^{-1}\, d\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\mathrm{diag}\left(\frac{1}{\omega^2 + A_{ii}^2}\right)\, d\omega \\
&= \frac{1}{2\pi}\pi(-A)^{-1} \\
&= -\frac{1}{2}A^{-1},
\end{aligned}$$

and plugging into $AQ + QA^{\mathsf{T}} + BB^{\mathsf{T}}$ gives,

$$A\left(-\frac{1}{2}A^{-1}\right) + \left(-\frac{1}{2}A^{-1}\right)A + I = 0$$

Now, examining the left hand side of the Parseval equality,

$$\mathrm{tr}\int_0^{\infty} H(t)H^{\mathsf{T}}(t)dt = \mathrm{tr}\int_0^{\infty} Ce^{At}BB^{\mathsf{T}}e^{A^{\mathsf{T}}t}C^{\mathsf{T}}\, dt = \mathrm{tr}\; CQC^{\mathsf{T}}\ .$$

And examining the right hand side of this equality,

$$\begin{aligned}
& \frac{1}{2\pi}\mathrm{tr}\int_{-\infty}^{\infty} H(\omega)H^*(\omega)\, d\omega \\
=\ & \frac{1}{2\pi}\mathrm{tr}\int_{-\infty}^{\infty} C(\omega I - A)^{-1}BB^{\mathsf{T}}(\omega I - A)^{-*}C^{\mathsf{T}}\, d\omega \\
=\ & \frac{1}{2\pi}\mathrm{tr}\; C\int_{-\infty}^{\infty}(\mathrm{i}\omega I - A)^{-1}BB^{\mathsf{T}}(\mathrm{i}\omega I - A)^{-*}\, d\omega\; C^{\mathsf{T}} \\
=\ & \mathrm{tr}\; CQC^{\mathsf{T}}
\end{aligned}$$

20.4 The H_2 norm of a system in terms of unit white noise response

A third interpretation of the H_2 norm of an LTI system is the sum of the variances of system responses $y(t)$ to uncorrelated unit white noise.

$$
\begin{aligned}
||H||_2^2 &= \lim_{t\to\infty} \mathsf{E}\ \mathrm{tr}\ [y^{\mathsf{T}}(t)y(t)] \qquad (79)\\
&= \lim_{t\to\infty} \mathsf{E}\ \mathrm{tr}\ [y(t)y^{\mathsf{T}}(t)] \left(= \lim_{t\to\infty} \mathsf{E}||y(t)y^{\mathsf{T}}(t)||_{\mathrm{F}}^2\right)\\
&= \lim_{t\to\infty} \mathsf{E}\ \mathrm{tr}\ [Cx(t)x^{\mathsf{T}}(t)C^{\mathsf{T}}]\\
&= \lim_{t\to\infty} \mathrm{tr}\ C\ \mathsf{E}\ [x(t)x^{\mathsf{T}}(t)]C^{\mathsf{T}}\\
&= \lim_{t\to\infty} \mathrm{tr}\ CQ(t)C^{\mathsf{T}}
\end{aligned}
$$

where $Q(t) = \mathsf{E}\ [x(t)x^{\mathsf{T}}(t)]$ is the (non-negative definite) state covariance matrix. Defining Q to be the limit of $Q(t)$ as t approaches infinity,

$$
||H||_2^2 = \mathrm{tr}\ CQC^{\mathsf{T}} \qquad (80)
$$

An r dimensional uncorrelated unit white noise process $u(t)$,

$$
u(t) = \begin{bmatrix} w_1(t) \\ \vdots \\ w_m(t) \end{bmatrix}
$$

has scalar components $w_i(t)$ having the following properties:

$$
\begin{aligned}
\mathsf{E}\ [w_i(t)] &= 0 \ \ \forall \ \ t && \text{... expected value}\\
\mathsf{E}\ [w_i(t_1)w_i(t_2)] &= \delta(t_1 - t_2) \ \ \forall \ \ t_1, t_2 && \text{... auto} - \text{correlation}\\
\mathsf{E}\ [w_i(t_1)w_j(t_2)] &= 0 \ \ \forall \ \ i, j, \ t_1, t_2 \ \ \text{and} \ \ i \neq j && \text{... cross} - \text{correlation}
\end{aligned}
$$

These properties, combined, result in facts that

$$
\mathsf{E}\ [u(t_1)u^{\mathsf{T}}(t_2)] = \delta(t_1 - t_2)I_r
$$

and that the power spectral density of *unit white* noise is 1 for *all* frequencies.

Now, examining the time-evolution of the state covariance matrix, we will see that the state covariance is the controllability gramian, in the limit as $t \to \infty$.

$$
\begin{aligned}
\dot{Q}(t) &= \frac{d}{dt}\mathsf{E}\left[x(t)x^{\mathsf{T}}(t)\right] \\
&= \mathsf{E}\left[\dot{x}(t)x^{\mathsf{T}}(t) + x(t)\dot{x}^{\mathsf{T}}(t)\right] \\
&= \mathsf{E}\left[(Ax+Bu)x^{\mathsf{T}}(t) + x(t)(Ax+Bu)^{\mathsf{T}}\right] \\
&= \mathsf{E}\left[Axx^{\mathsf{T}} + xx^{\mathsf{T}}A^{\mathsf{T}}\right] + \mathsf{E}\left[Bux^{\mathsf{T}} + xu^{\mathsf{T}}B^{\mathsf{T}}\right] \\
&= AQ(t) + Q(t)A^{\mathsf{T}} + \\
&\quad \mathsf{E}\left[Bu(t)\left(e^{At}x_o + \int_0^t e^{A(t-\tau)}Bu(\tau)\; d\tau\right)^{\mathsf{T}} + \left(e^{At}x_o + \int_0^t e^{A(t-\tau)}Bu(\tau)\; d\tau\right)\; u^{\mathsf{T}}B^{\mathsf{T}}\right] \\
&= AQ(t) + Q(t)A^{\mathsf{T}} + \\
&\quad \mathsf{E}\left[B\int_0^t u(t)u^{\mathsf{T}}(\tau)B^{\mathsf{T}}e^{A^{\mathsf{T}}(t-\tau)}\; d\tau + \int_0^t e^{A(t-\tau)}Bu(\tau)u^{\mathsf{T}}(t)\; d\tau B^{\mathsf{T}}\right] \\
&= AQ(t) + Q(t)A^{\mathsf{T}} + \\
&\quad B\int_0^t \delta(t-\tau)B^{\mathsf{T}}e^{A^{\mathsf{T}}(t-\tau)}\; d\tau + \int_0^t e^{A(t-\tau)}B\delta(t-\tau)\; d\tau B^{\mathsf{T}} \\
&= AQ(t) + Q(t)A^{\mathsf{T}} + \frac{1}{2}BB^{\mathsf{T}} + \frac{1}{2}BB^{\mathsf{T}} \\
&= AQ(t) + Q(t)A^{\mathsf{T}} + BB^{\mathsf{T}} \qquad (81)
\end{aligned}
$$

the solution to this differential equation is

$$Q(t) = e^{At}Q(0)e^{A^{\mathsf{T}}t} + \int_0^t e^{A\tau}BB^{\mathsf{T}}e^{A^{\mathsf{T}}\tau}\; d\tau$$

where the initial state covariance matrix $Q(0)$, is

$$Q(0) = \mathsf{E}\left[x(0)x^{\mathsf{T}}(0)\right]$$

For asymptotically stable systems, e^{At} approaches zero as t approaches infinity, and the transient response $e^{At}Q(0)e^{A^{\mathsf{T}}t}$ also approaches zero. Hence

$$\lim_{t\to\infty} Q(t) = \int_0^t e^{A\tau}BB^{\mathsf{T}}e^{A^{\mathsf{T}}\tau}\; d\tau$$

which is the definition of the controllability gramian. Equivalently, for asymptotically stable systems,

$$\lim_{t\to\infty} \dot{Q}(t) = 0 = AQ + QA^{\mathsf{T}} + BB^{\mathsf{T}}$$

and Q is the controllability gramian.

Using the property of the Frobeneus norm that $||A||_{\mathrm{F}} = ||A^{\mathsf{T}}||_{\mathrm{F}}$, the two interpretations of the H_2 norm of of an LTI system presented in the preceding pages (the interpretation in terms of white noise and the interpretation in terms of unit impulse responses) can be shown to be equivalent.

$$||H||_2^2 = \mathrm{tr}\ B^{\mathsf{T}} P B = \mathrm{tr}\ C Q C^{\mathsf{T}}\ ,$$

where

$$0 = A^{\mathsf{T}} P + P A + C^{\mathsf{T}} C \qquad \text{and} \qquad 0 = A Q + Q A^{\mathsf{T}} + B B^{\mathsf{T}}$$

Proof 1:

$$\begin{aligned} ||H||_2^2 &= \int_0^\infty ||H(t)||_{\mathrm{F}}^2\ dt = \mathrm{tr}\ \int_0^\infty (C e^{At} B)^{\mathsf{T}} (C e^{At} B)\ dt \\ &= \mathrm{tr}\ B^{\mathsf{T}} \int_0^\infty e^{A^{\mathsf{T}} t} C^{\mathsf{T}} C e^{At}\ dt\ B = \mathrm{tr}\ B^{\mathsf{T}} P B \\ &= \mathrm{tr}\ C \int_0^\infty e^{At} B B^{\mathsf{T}} e^{A^{\mathsf{T}} t}\ dt\ C^{\mathsf{T}} = \mathrm{tr}\ C Q C^{\mathsf{T}} \end{aligned}$$

Proof 2:

$$\begin{aligned} \mathrm{tr}(B^{\mathsf{T}} P B) &= \mathrm{tr}(B B^{\mathsf{T}} P) \\ &= \mathrm{tr}((-AQ - QA^{\mathsf{T}})P) \\ &= \mathrm{tr}(-AQP - QA^{\mathsf{T}} P) \\ &= \mathrm{tr}(-QPA - QA^{\mathsf{T}} P) \\ &= \mathrm{tr}(Q C^{\mathsf{T}} C) \\ &= \mathrm{tr}(C^{\mathsf{T}} Q C) \end{aligned}$$

In summary, the H_2 norm of a system can be equivalently interpreted as:

- the definite integral (from $t = 0$ to ∞) of the sum of the squares of the unit impulse responses,
- the sum of the areas under the magnitude-squared frequency response functions, and
- the expected value of the sum of the squares of the outputs responding to uncorrelated unit white noise.

These norms can be calculated by solving Liapunov equations for the controllability gramian and the observability gramian.

Note, finally, that the H_2 norm of a continuous-time LTI system with $D \neq 0$ does not exist.

Continuous-time LTI systems with $D = 0$ are called *strictly proper*.

21 Feedback Control for Stabilization

If the inputs $u(t)$ to a dynamic system $\dot{x} = Ax + Bu$ are linearly related to the states by a constant *full state feedback gain matrix*, $u(t) = -Kx(t)$, then the *closed loop* system $\dot{x} = (A - BK)x$ is autonomous. If the the pair (A, B) is controllable, then a full state feedback gain matrix K may be found that places the eigenvalues of $(A - BK)$ to any desired values.

22 Feedfoward Control for Tracking

If the inputs $u(t)$ to a stable dynamic system $\dot{x} = Ax + Bu$ are linearly related to desired output values $r(t)$ by a constant and square *feedforward gain matrix*, $u(t) = Fr(t)$, then the *feedforward dynamics* are $\dot{x} = Ax + BFr$. and outputs $y(t) = Cx(t)$ can be made to asymptotically approach $r(t)$ by setting $R = Y(\infty) = -CA^{-1}BFR$ from which $F = (-CA^{-1}B)^{-1}$.

23 Feedback Control for Stabilization and Tracking

If certain outputs $y(t) = Cx(t)$ are to track a set of inputs $r(t)$, the rate of the integrated tracking error is $\dot{q}(t) = y(t) - r(t)$. Augmenting the system dynamics with the dynamics of the tracking error,

$$\begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ -I \end{bmatrix} r + Bu$$

Setting the controls u to be linear in the states and the tracking error,
$u = -K[x^{\mathsf{T}}, q^{\mathsf{T}}]^{\mathsf{T}}$,

$$\begin{bmatrix} \dot{x} \\ \dot{q} \end{bmatrix} = \left[\begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} - BK \right] \begin{bmatrix} x \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ -I \end{bmatrix} r$$

the states and the tracking error can be selectively stabilized to zero by appropriate selection of the feedback gain matrix K as long as the augmented system is controllable through B. This approach does not require the number of tracked outputs to equal the number of inputs.

24 Transforming Continuous Time Systems to Discrete-Time Systems (ZOH)

Consider the evolution of the state response from a time t to a time $t+\Delta t$ for inputs $u(\tau)$ that equal $u(t)$ for $t \le \tau \le t+\Delta t$. This is a *zero-order hold* (ZOH) on $u(t)$ over the interval $[t, t+\Delta t]$. The initial condition to this evolution is $x(t)$, and we wish to find the states $x(t+\Delta t)$. Shifting time to be 0 at time t in equation (35) and defining $x(k\Delta t) \equiv x(k)$, $x((k+1)\Delta t) \equiv x(k+1)$, and, $u(\tau) \equiv u(k\Delta t) \equiv u(k)$. for $0 \le \tau \le \Delta t$, gives

$$\begin{aligned} x(k+1) &= e^{A\Delta t} x(k) + \int_0^{\Delta t} e^{A(\Delta t - \tau)} \, d\tau \; Bu(k) \\ &= \left[e^{A\Delta t}\right] x(k) + \left[A^{-1}\left(e^{A\Delta t} - I\right) \; B\right] u(k) \; , \\ &= [A_{\mathsf{d}}] \; x(k) + [B_{\mathsf{d}}] \; u(k) \end{aligned} \tag{82}$$

where A_{d} and B_{d} are the *discrete-time* dynamics matrix and the *discrete-time* input matrix, $A_{\mathsf{d}} = e^{A\Delta t}$ and $B_{\mathsf{d}} = A^{-1}(A_{\mathsf{d}} - I)B$.

Note that B_{d} exists even though A may not be invertible. Consider the diagonalization $A = \bar{X}^{-1}\Lambda\bar{X}$. The inverse of A may be expressed $A^{-1} = \bar{X}^{-1}\Lambda^{-1}\bar{X}$, where $\bar{X}\bar{X}^{-1} = I$. So,

$$A^{-1}(e^{At} - I) = \bar{X}^{-1}\Lambda^{-1}\bar{X}\bar{X}^{-1}(e^{\Lambda t} - I)\bar{X} = \bar{X}^{-1}\Lambda^{-1}(e^{\Lambda t} - I)\bar{X},$$

which contains diagonal terms $(e^{\lambda_i} - 1)/\lambda_i$, and

$$\lim_{\lambda_i \to 0} \frac{e^{\lambda_i} - 1}{\lambda_i} = 1 \; .$$

To compute B_{d} without inverting A, note the following:

$$\begin{aligned} e^{A\Delta t} - I &= A\Delta t + \frac{\Delta t^2}{2} AA + \frac{\Delta t^3}{6} AAA + \frac{\Delta t^4}{24} AAAA + \cdots \\ A^{-1}(e^{A\Delta t} - I) &= \Delta t + \frac{\Delta t^2}{2} A + \frac{\Delta t^3}{6} AA + \frac{\Delta t^4}{24} AAA + \cdots \\ B_{\mathsf{d}} = A^{-1}(e^{A\Delta t} - I)B &= \Delta t B + \frac{\Delta t^2}{2} AB + \frac{\Delta t^3}{6} AAB + \frac{\Delta t^4}{24} AAAB + \cdots \; , \end{aligned}$$

and, if

$$M = \begin{bmatrix} A_{n\times n} & B_{n\times r} \\ 0_{r\times n} & 0_{r\times r} \end{bmatrix} \; , \tag{83}$$

then,

$$\begin{aligned} e^{M\Delta t} &= I + \Delta t M + \frac{\Delta t^2}{2} MM + \frac{\Delta t^3}{6} MMM + \frac{\Delta t^4}{24} MMMM + \cdots \\ &= I + \Delta t \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} + \frac{\Delta t^2}{2} \begin{bmatrix} AA & AB \\ 0 & 0 \end{bmatrix} + \frac{\Delta t^3}{6} \begin{bmatrix} AAA & AAB \\ 0 & 0 \end{bmatrix} + \cdots \\ &= \begin{bmatrix} I_n + \Delta t A + \frac{\Delta t^2}{2} AA + \cdots & \Delta t B + \frac{\Delta t^2}{2} AB + \frac{\Delta t^3}{6} AAB + \cdots \\ 0 & I_r \end{bmatrix} , \end{aligned}$$

so

$$e^{M\Delta t} = \begin{bmatrix} A_{\mathsf{d}} & B_{\mathsf{d}} \\ 0 & I_r \end{bmatrix} , \tag{84}$$

and the discrete-time dynamics matrix A_{d} and input matrix B_{d} may be computed using a single matrix exponential computation.

In MATLAB:

```
[n,r] = size(B);
M = [ A B ; zeros(r,n+r) ];
eMdt = expm(M*dt);
Ad = eMdt(1:n,1:n);
Bd = eMdt(1:n,n+1:n+r);
```

With the matrices A_{d} and B_{d}, the transient response may be computed digitally using equation (82).

```
x(:,1) = x0;
for p=1:points-1
    x(:,p+1) = Ad * x(:,p) + Bd * u(:,p);
end
```

25 Transforming Continuous Time Systems to Discrete-Time Systems (FOH)

Using the same time-shifting as in the previous ZOH derivation, but now specifying that the input changes *linearly* over a time increment, Δt,

$$u(\tau) = u(k) + \frac{1}{\Delta t}(u(k+1) - u(k))\ \tau \equiv u(k) + u'(k)\ \tau \ \text{ for } \ 0 \le \tau \le \Delta t \ , \tag{85}$$

This is a *first order hold* (FOH) on the inputs over $[t, t + \Delta t]$. Defining $x(k\Delta t) \equiv x(k)$, $u(k\Delta t) = u(k)$ gives

$$\begin{aligned}
x(k+1) &= e^{A\Delta t} x(k) + \int_0^{\Delta t} e^{A(\Delta t - \tau)}\ B\ u(\tau)\ d\tau \\
&= e^{A\Delta t} x(k) + \int_0^{\Delta t} e^{A(\Delta t - \tau)}\ d\tau\ B\ u(k) + \int_0^{\Delta t} e^{A(\Delta t - \tau)}\ B\ u'(k)\ \tau\ d\tau \\
&= e^{A\Delta t} x(k) + A^{-1}\left(e^{A\Delta t} - I\right)\ B\ u(k) + A^{-2}\left(e^{A\Delta t} - I - A\Delta t\right)\ B\ u'(k) \\
&= \left[e^{A\Delta t}\right]\ x(k) + \left[A^{-1}\left(e^{A\Delta t} - I\right)\ B\right]\ u(k) + \left[A^{-2}\left(e^{A\Delta t} - I - A\Delta t\right)\ B\right]\ u'(k)\ , \\
&= [A_{\mathsf{d}}]\ x(k) + [B_{\mathsf{d}}]\ u(k) + [B'_{\mathsf{d}}]\ u'(k)
\end{aligned} \tag{86}$$

where A_{d}, B_{d}, and B'_{d} are the discrete-time dynamics matrix and the discrete-time input matrices.

The input matrices B_{d} and B'_{d} exist even though A may not be invertible. To compute B_{d} without inverting A, note the following:

$$\begin{aligned}
e^{A\Delta t} - I &= A\Delta t + \frac{\Delta t^2}{2} AA + \frac{\Delta t^3}{6} AAA + \frac{\Delta t^4}{24} AAAA + \cdots \\
A^{-1}(e^{A\Delta t} - I) &= \Delta t + \frac{\Delta t^2}{2} A + \frac{\Delta t^3}{6} AA + \frac{\Delta t^4}{24} AAA + \cdots \\
B_{\mathsf{d}} = A^{-1}(e^{A\Delta t} - I)B &= \Delta t B + \frac{\Delta t^2}{2} AB + \frac{\Delta t^3}{6} AAB + \frac{\Delta t^4}{24} AAAB + \cdots \ ,
\end{aligned}$$

and,

$$\begin{aligned}
e^{A\Delta t} - I - A\Delta t &= \frac{\Delta t^2}{2} AA + \frac{\Delta t^3}{6} AAA + \frac{\Delta t^4}{24} AAAA + \cdots \\
A^{-2}(e^{A\Delta t} - I - A\Delta t) &= \frac{\Delta t^2}{2} + \frac{\Delta t^3}{6} A + \frac{\Delta t^4}{24} AA + \cdots \\
B'_{\mathsf{d}} = A^{-2}(e^{A\Delta t} - I - A\Delta t)B &= \frac{\Delta t^2}{2} B + \frac{\Delta t^3}{6} AB + \frac{\Delta t^4}{24} AAB + \cdots \ ,
\end{aligned}$$

If

$$M \triangleq \begin{bmatrix} A_{n\times n} & B_{n\times r} & 0_{n\times r} \\ 0_{r\times n} & 0_{r\times r} & I_{r\times r} \\ 0_{r\times n} & 0_{r\times r} & 0_{r\times r} \end{bmatrix}_{(n+2r)\times(n+2r)} , \tag{87}$$

then,

$$\begin{aligned} e^{M\Delta t} &= I + \Delta t M + \frac{\Delta t^2}{2} MM + \frac{\Delta t^3}{6} MMM + \frac{\Delta t^4}{24} MMMM + \cdots \\ &= I + \Delta t \begin{bmatrix} A & B & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} + \frac{\Delta t^2}{2} \begin{bmatrix} AA & AB & B \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{\Delta t^3}{6} \begin{bmatrix} AAA & AAB & AB \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \cdots \\ &= \begin{bmatrix} I_n + \Delta t A + \frac{\Delta t^2}{2} AA + \cdots & \Delta t B + \frac{\Delta t^2}{2} AB + \frac{\Delta t^3}{6} AAB + \cdots & \frac{\Delta t^2}{2} B + \frac{\Delta t^3}{6} AB + \cdots \\ 0 & I_r & \Delta t \; I_r \\ 0 & 0 & I_r \end{bmatrix} , \end{aligned}$$

so,

$$e^{M\Delta t} = \begin{bmatrix} A_{\mathsf{d}} & B_{\mathsf{d}} & B'_{\mathsf{d}} \\ 0 & I_r & \Delta t \; I_r \\ 0 & 0 & I_r \end{bmatrix} , \tag{88}$$

and the discrete-time dynamics matrix A_{d} and input matrices B_{d} and B'_{d} may be computed using a single matrix exponential computation.

The discrete-time system with ramp inputs between sample points is then

$$\begin{aligned} x(k+1) &= A_{\mathsf{d}} \; x(k) + [B_{\mathsf{d}} - B'_{\mathsf{d}}/(\Delta t)] \; u(k) + [B'_{\mathsf{d}}/(\Delta t)] \; u(k+1) \\ &= A_{\mathsf{d}} \; x(k) + B_{\mathsf{d0}} \; u(k) - B_{\mathsf{d1}} \; u(k+1) & (89) \\ y(k) &= C \; x(k) + D \; u(k) & (90) \end{aligned}$$

In this system, $y(k+1)$ depends on $x(k+1)$, which in turn depends on $u(k+1)$. So even if $D = 0$, $u(k+1)$ feeds through to $y(k+1)$. This discrete-time system can be expressed in a new state, $\bar{x}(k) = x(k) - B_{\mathsf{d1}} u(k)$, giving

$$\begin{aligned} \bar{x}(k+1) &= A_{\mathsf{d}} \; \bar{x}(k) + \bar{B}_{\mathsf{d}} \; u(k) \\ y(k) &= C \; \bar{x}(k) + \bar{D} \; u(k) \end{aligned} \tag{91}$$

where $\bar{B}_{\mathsf{d}} = B_{\mathsf{d0}} + A_{\mathsf{d}} B_{\mathsf{d1}}$ and $\bar{D} = D + C B_{\mathsf{d1}}$.

26 Discrete-Time Linear Time Invariant Systems

The transformation from continuous-time to discrete-time models provides a convenient method for digital simulation of transient system responses. In addition, the dynamics of discrete-time systems may be investigated analytically. This section summarizes analytical solutions to discrete-time LTI systems described by

$$\begin{aligned} x(k+1) &= A\, x(k) + B\, u(k) \\ y(k) &= C\, x(k) + D\, u(k) \end{aligned} \tag{92}$$

Discrete-time models imply that the frequencies of harmonic components of dynamic variables are limited to the *Nyquist interval*, $\omega \in [-\pi/(\Delta t), +\pi/(\Delta t)]$. Harmonic components with frequencies outside this range are *aliased* into the Nyquist interval, when they are sampled. Frequency components of analog signals outside the Nyquist range should be filtered-out with so-called (low-pass) anti-alias filters prior to sampling.

Hereinafter, the discrete-time dynamics matrix and input matrix are denoted A and B. Expressions in this section involving the continuous-time dynamics and input matrices will have a subscript c, as in A_c, B_c, and λ_c.

27 Band limited signals and aliasing

The frequencies contained in continuous time signals can be arbitrarily high. (Electromagnetic waves are in the GHz range.) Frequencies present in discrete-time signals are limited to within the *Nyquist* frequency range. Consider a continuous time signal of duration T sampled at N points uniformly spaced at an interval Δt, $T = N(\Delta t)$. The longest period (lowest frequency) fully contained in a signal of duration T has a period of T and a frequency of $1/T$. (The lowest (non-zero) frequency f_1 contained in a signal of duration T is $1/T$.) This lowest frequency value is the frequency increment Δf of the sampled signal's discrete Fourier transform. The highest frequency $f_{\max}$ in the discrete Fourier transform is $(N/2)(\Delta f)$. So, substituting,

$$f_{\max} = \frac{N}{2}(\Delta f) = \frac{N}{2}\frac{1}{T} = \frac{N}{2}\frac{1}{N(\Delta t)} = \frac{1}{2(\Delta t)} \equiv f_{\mathrm{N}} \tag{93}$$

The frequency content of discrete time signals are limited to the Nyquist frequency range, $-f_{\rm N} < f \leq +f_{\rm N}$.

In continuous time, unit white noise has a power spectral density of 1 for *all* frequencies, and an auto-correlation function of $R(\tau) = \delta(\tau)$. In discrete time, unit white noise has a power spectral density of 1 over the Nyquist frequency band, and an auto-correlation function equivalent to the sinc function.

$$S(f) = \begin{cases} 1 & -f_{\rm N} < f < f_{\rm N} \\ 0 & f < -f_{\rm N}, f_{\rm N} < f \end{cases} \tag{94}$$

$$R(\tau) = \frac{1}{\pi f_{\rm N} \tau} \sin(2\pi f_{\rm N} \tau) \tag{95}$$

According to Parseval's Theorem, the mean square of unit white noise in discrete time is $\sigma^2 = 2f_{\rm N} = 2/(2(\Delta t)) = 1/(\Delta t)$ and the root mean square of a unit white noise process is $1/\sqrt{(\Delta t)}$.

If a continuous time signal is sampled at a sampling rate $(1/(\Delta t))$ which is lower than twice the highest frequency components present in the continuous time signal, the sampled signal will appear in the discrete time sequence as an *aliased component*, that is, at a frequency less than the Nyquist frequency, as shown in Figure 2.

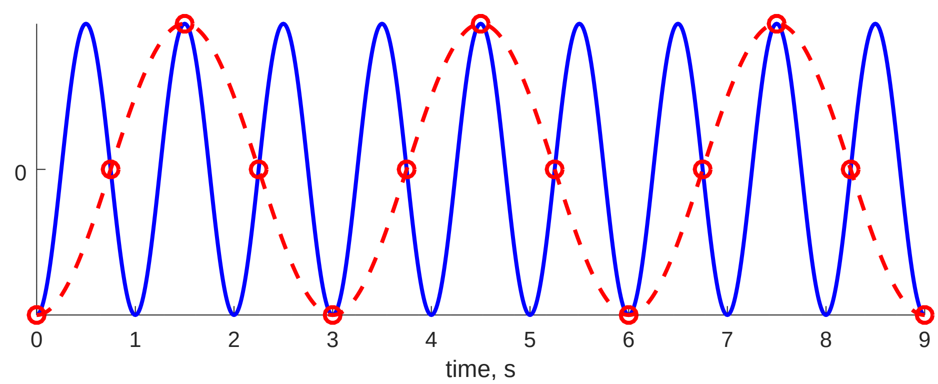

Figure 2. A 1 Hz continuous time sinusoid (blue) sampled at $(\Delta t) = 0.75$ s (red points) appears as a signal with a three-second period (1/3 Hz) (red line).

A relation between the continuous time frequency and its aliased frequency may be derived by thinking of a frequency value as the sum of an integer component and a fractional component. This derivation makes use of two trigonometric identities. For an integer n, $n \in 0, 1, 2, ...$ and a rational number r, $0 < r < 1$,

$$\cos(2\pi(n + r) + \phi) = \cos(2\pi r + \phi) \tag{96}$$

and for any values of r and ϕ,

$$\cos(2\pi r + \phi) = \cos(-2\pi r - \phi + 2\pi) = \cos(2\pi(1 - r) - \phi) \tag{97}$$

With these identities,

$$y(k) = \cos(2\pi f k(\Delta t) + \phi) = \cos(2\pi k((f\Delta t)_n + (f\Delta t)_r) + \phi) \tag{98}$$

where $(f\Delta t)_n$ is an integer and $(f\Delta t)_r$ is a rational (remainder) between 0 and 1. So,

$$\begin{aligned} y(k) = \cos(2\pi f k\Delta t + \phi) &= \cos(2\pi k(f\Delta t)_r + \phi) && (99)\\ &= \cos(2\pi k(1 - (f\Delta t)_r) - \phi) && (100)\\ &= \cos(2\pi k f_{\mathrm{a}} \Delta t - \phi) && (101) \end{aligned}$$

and the frequency of the aliased signal, f_{a} is

$$f_{\mathrm{a}} = \begin{cases} (f\Delta t)_r/(\Delta t) & \text{if} \quad (f\Delta t)_r < 1/2 \\ (1 - (f\Delta t)_r)/(\Delta t) & \text{if} \quad (f\Delta t)_r > 1/2 \end{cases} \tag{102}$$

The power spectral density of an aliased signal $S_{\mathrm{a}}(f)$ is related to its continuous-time power spectral density $S(f)$ as

$$S_{\mathrm{a}}(f) = S(f) + \sum_{k=1}^{\infty} S\left(k/(\Delta t) - f\right) + S\left(k/(\Delta t) + f\right) \tag{103}$$

where $0 \leq f\Delta t \leq 1/2$. Figure 3 shows how the power spectral density of an aliased signal involves an accordion-like wrapping of the frequency components outside of the Nyquist frequency range into the Nyquist bandwidth.

Once higher frequency components have been aliased into the Nyquist band, it is impossible to determine if a peak in the power spectral density has been aliased from a higher frequency or not. For this reason, signals should be sampled at a frequency that is ten or more times the highest frequency of interest, and should be anti-alias filtered at the Nyquist frequency, or at a frequency slightly lower than the Nyquist frequency. Note that:

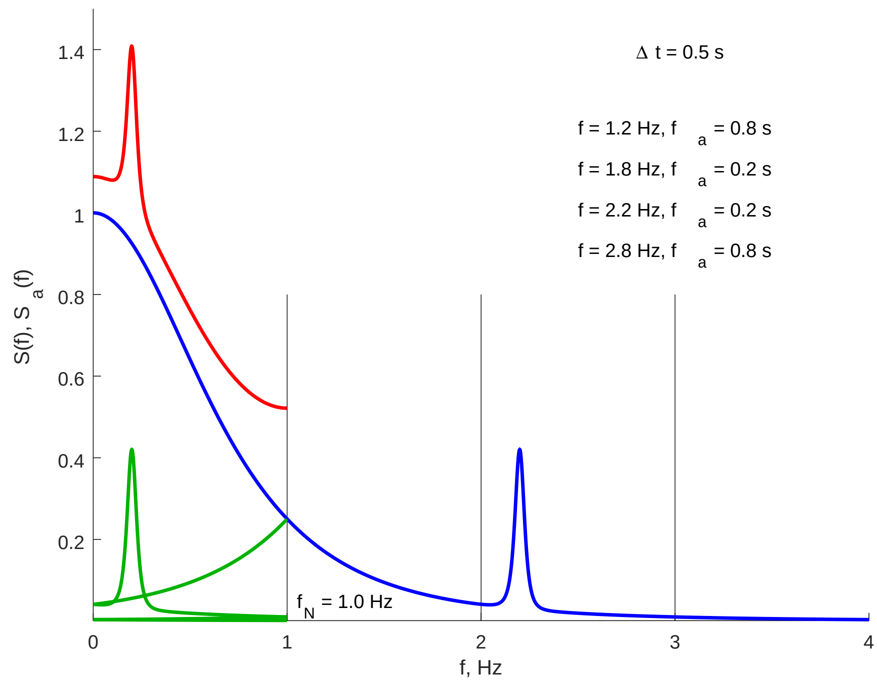

Figure 3. The power spectral density of a continuous time signal with a spectral peak at 2.2 Hz (blue). The signal sampled at $\Delta t = 0.5$ s is aliased to a signal with a spectral peak at 0.2 Hz (red).

- Sampling a continuous time signal without anti-alias filtering, concentrates all of the signal energy into the Nyquist frequency range. The mean square of a signal sampled without filtering equals the mean square of the continuous time signal.

- The mean square of a sampled low-pass filtered signal is always less than the mean square of the continuous time signal.

- Anti-alias filtering introduces delays and potentially phase distortion in the signal. Matched linear-phase anti-alias filters are generally preferred for this purpose. "Sigma-Delta" analog-to-digital converters inherently incorporate anti-alias filtering.

28 State Response Sequence

Applying the dynamics equation of (92) recursively from a known initial state $x(0)$ and with a known input sequence,

$$\begin{aligned} x(1) &= Ax(0) + Bu(0) \\ x(2) &= A^2x(0) + ABu(0) + Bu(1) \\ x(3) &= A^3x(0) + A^2Bu(0) + ABu(1) + Bu(2) \\ &\vdots \\ x(k) &= A^kx(0) + A^{k-1}Bu(0) + \cdots + ABu(k-2) + Bu(k-1) \end{aligned} \tag{104}$$

or, starting at any particular time step, $j > 0$ and advancing by k time steps,

$$x(j+k) = A^kx(j) + \sum_{i=1}^{k} A^{i-1}Bu(k+j-i) \tag{105}$$

$$y(j+k) = CA^kx(j) + \sum_{i=1}^{k} CA^{i-1}Bu(j+k-i) + Du(j+k) \tag{106}$$

The first terms in (105) and (106) are the free state responses of the state and the output (the homogeneous solution) to the difference equations; the second terms are the forced responses (the particular solution).

29 Eigenvalues and Diagonalization in Discrete-Time

The discrete-time system may be diagonalized with the eigenvector matrix of the dynamics matrix.

$$\bar{X}^{-1}A\bar{X} = \Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}$$

where Λ is the diagonal matrix of discrete-time eigenvalues and the eigenvalue problem is

$$A\bar{X} = \bar{X}\Lambda \ .$$

Note that the continuous-time dynamics matrix A_c and the discrete-time dynamics matrix A have the same eigenvectors $\bar{X}$, and that the eigenvalues of the discrete-time dynamics matrix are related to the eigenvalues of the continuous-time dynamics matrix through the scalar exponential

$$\exp[\lambda_{\mathsf{c}i}\Delta t] = \lambda_i .$$

The diagonalized system (in modal coordinates) is

$$\begin{array}{rcl} q(k+1) & = & \Lambda\ q(k)\ +\ \bar{X}^{-1}B\ u(k) \\ y(k) & = & C\bar{X}\ q(k) + D\ u(k) \end{array} \tag{107}$$

and the modal response sequence from time step j to time step $j+k$ is

$$q(j+k) = \Lambda^k q(j) + \sum_{i=1}^{k} \Lambda^{i-1}\bar{X}^{-1}B\ u(j+k-i) \tag{108}$$

The stability of a mode of a discrete-time system is determined from the magnitude of the eigenvalue. Considering the free responses of (107) or (108), if $|\lambda_i| > 1$ then $q_i(k)$ will grow exponentially with k.

$$\begin{array}{rcccl} \text{mode } i \text{ is stable:} & \Leftrightarrow & |\lambda_i| < 1 & \Leftrightarrow & \mathsf{Re}(\lambda_{\mathsf{c}i}) < 0 \\ \text{mode } i \text{ is unstable:} & \Leftrightarrow & |\lambda_i| > 1 & \Leftrightarrow & \mathsf{Re}(\lambda_{\mathsf{c}i}) > 0 \end{array}$$

A system is stable if and only if all of its modes are stable.

Note that the elements of the modal sequence vector $q(k+j)$ may be evaluated individually, since Λ^j is diagonal. For systems with under-damped dynamics the continuous-time and discrete-time eigenvalues are complex-valued. The eigenvalues and modal coordinates appear in complex-conjugate pairs. The sum of the complex conjugate modal coordinates $q(k) + q^*(k)$ is *real* valued and equals twice the real part of $q(k)$.

30 Discrete-time convolution and Unit Impulse Response Sequence (Markov Parameters and Hankel Matrices)

Applying equations (92) and (104) to derive the output sequence $y(k)$ from $x(0) = 0$ and $u(k) = 0 \; \forall \; k < 0$,

$$\begin{aligned} y(0) &= Du(0) \\ y(1) &= CBu(0) + Du(1) \\ y(2) &= CABu(0) + CBu(1) + Du(2) \\ y(3) &= CA^2Bu(0) + CABu(1) + CBu(2) + Du(3) \\ &\vdots \\ y(k) &= \sum_{i=1}^{k} CA^{i-1}B \; u(k-i) + Du(k) \end{aligned} \tag{109}$$

This is a discrete-time convolution, and it may be re-expressed as either

$$y(k) = \sum_{i=0}^{k} Y(i) \; u(k-i) \tag{110}$$

or, by substituting $k - i = j$ and noting $i = 0 \Leftrightarrow j = k$ and $i = k \Leftrightarrow j = 0$,

$$y(k) = \sum_{j=0}^{k} Y(k-j) \; u(j) \tag{111}$$

where the kernel terms of this convolution are called *Markov parameters*:

$$\begin{aligned} Y &= \left[\; D \; , \; CB \; , \; CAB \; , \; CA^2B \; , \; \cdots \; , \; CA^{k-1}B \; \right]_{m \times r(k+1)} \\ Y(0) &= D \; , \quad Y(i) = CA^{i-1}B \quad \forall \quad i > 0 \; , \quad Y(i) \in \mathbb{R}^{m \times r} \end{aligned} \tag{112}$$

The first Markov parameter, $Y(0)$, is the feed-through matrix, D. The output evolving from a zero initial state and a unit impulse input ($u(0) = I_r$, $u(k) = 0$ for $k \neq 0$) is the sequence of Markov parameters.

The scalar-valued sequence of the (p, q) terms of the sequence of matrix-valued Markov parameters,

$$\frac{1}{\Delta t}[Y(0)]_{p,q} \; , \; \frac{1}{\Delta t}[Y(1)]_{p,q} \; , \; \frac{1}{\Delta t}[Y(2)]_{p,q} \; , \; \cdots$$

is the unit impulse response of element p from a unit impulse on element q. The unit impulse sequence is $[1/(\Delta t), \; 0, \cdots , \; 0]$

For asymptotically stable (AS) discrete-time LTI systems, free responses decay asymptotically to zero. So, in principle, the sequence of Markov parameters is infinitely long. But, note that for $x(0) = 0$ and $u(i) = 0 \ \forall \ i < 0$,

$$y(k) = \sum_{i=0}^{\infty} Y(i) \ u(k-i) = \sum_{i=0}^{k} Y(i) \ u(k-i) \tag{113}$$

A forced response sequence evolving from $x(0) = 0$ with $u(i) = 0 \ \forall \ i < 0$, is linear in the Markov parameters.

$$\begin{aligned} &\left[\ y(0) \ \ y(1) \ \ y(2) \ \ y(3) \ \ \cdots \ \ y(j)\ \right]_{m \times (j+1)} = \\ &\left[\ D \ \ CB \ \ CAB \ \ CA^2B \ \ \cdots \ \ CA^{j-1}B\ \right]_{m \times r(j+1)} \cdot \\ &\begin{bmatrix} u(0) & u(1) & u(2) & u(3) & \cdots & u(j) \\ & u(0) & u(1) & u(2) & \cdots & u(j-1) \\ & & u(0) & u(1) & \cdots & u(j-2) \\ & & & u(0) & \cdots & u(j-3) \\ & & & & \ddots & \vdots \\ & & & & & u(0) \end{bmatrix}_{r(j+1) \times (j+1)} \end{aligned} \tag{114}$$

The forced response sequence of a system evolving from $x(0) = 0$ with an (assumed) finite sequence of $p + 1$ Markov parameters is

$$y(k) = \sum_{i=0}^{p} Y(i) \ u(k-i)$$

or

$$\left[\begin{array}{cccccc} y(p) & y(p+1) & y(p+2) & y(p+3) & \cdots & y(p+j) \end{array}\right]_{m\times(j+1)} =$$

$$\left[\begin{array}{ccccccc} CA^{p-1}B & CA^{p-2}B & \cdots & CA^2B & CAB & CB & D \end{array}\right]_{m\times r(p+1)} \cdot$$

$$\left[\begin{array}{cccccc} u(0) & u(1) & u(2) & u(3) & \cdots & u(j) \\ u(1) & u(2) & u(3) & u(4) & \cdots & u(j+1) \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ u(p-3) & u(p-2) & u(p-1) & u(p) & \cdots & u(p+j-3) \\ u(p-2) & u(p-1) & u(p) & u(p+1) & \cdots & u(p+j-2) \\ u(p-1) & u(p) & u(p+1) & u(p+2) & \cdots & u(p+j-1) \\ u(p) & u(p+1) & u(p+2) & u(p+3) & \cdots & u(p+j) \end{array}\right]_{r(p+1)\times(j+1)} \quad (115)$$

The matrix built of the input sequence $[u(0), \cdots, u(p+j)]$ is called a *block Hankel matrix* of the input seqence. The output sequence $[y(p), \cdots, y(p+j)]$ is a linear combination of the rows of the input sequence Hankel matrix. In other words, the rows of the Hankel matrix form a linear basis for the output sequence $[y(p), \cdots, y(p+j)]$.

$$\left[\begin{array}{cccccc} y(p) & y(p+1) & y(p+2) & y(p+3) & \cdots & y(p+j) \end{array}\right]_{m\times(j+1)} =$$

$$\begin{array}{rl}
Y(p) & \left[\begin{array}{cccccc} u(0) & u(1) & u(2) & u(3) & \cdots & u(j) \end{array}\right]_{r\times(j+1)} + \\
Y(p-1) & \left[\begin{array}{cccccc} u(1) & u(2) & u(3) & u(4) & \cdots & u(j+1) \end{array}\right]_{r\times(j+1)} + \\
& \vdots \\
Y(3) & \left[\begin{array}{cccccc} u(p-3) & u(p-2) & u(p-1) & u(p) & \cdots & u(p+j-3) \end{array}\right] + \\
Y(2) & \left[\begin{array}{cccccc} u(p-2) & u(p-1) & u(p) & u(p+1) & \cdots & u(p+j-2) \end{array}\right] + \\
Y(1) & \left[\begin{array}{cccccc} u(p-1) & u(p) & u(p+1) & u(p+2) & \cdots & u(p+j-1) \end{array}\right] + \\
Y(0) & \left[\begin{array}{cccccc} u(p) & u(p+1) & u(p+2) & u(p+3) & \cdots & u(p+j) \end{array}\right]_{r\times(j+1)}
\end{array}$$

Note that in equation (115) the sequence of Markov parameters is given in reverse order. Given an assumption for the impulse response duration p and input/output sequences $[u(0), \cdots, u(p+j)]$ and $[y(p), \cdots y(p+j)]$, equation

(115) provides a linear model for the sequence of $p+1$ Markov parameters, $Y(i)$, $i = p, \cdots, 0$. In this estimation problem there are $m \times (j+1)$ equations and $m \times r(p+1)$ unknown model coefficients. So the length of the data sequence, j, should be much larger than the number of inputs r times the impulse response duration p. The estimation of Markov parameters is a time-domain approach to Wiener filtering and is the first step in the Eigensystem Realization Algorithm (ERA) for identification of state-space models.

If we "know" the MIMO system to be strictly proper ($D = 0_{(m \times r)}$) then the model may be expressed without the feedthrough matrix ($Y(0)$) by removing D from the set of Markov parameters and removing the first r rows from the matrix of input data.

31 An example of a discrete time system: the running average

Consider a sequence of numbers as the input to a discrete-time LTI system,

$$u(0),\ u(1),\ u(2),\ u(3),\ u(4),\ \ldots$$

A *running average* of this sequence can be defined as the weighted arithmetic average of the previous running average, $\bar{u}(k)$, and the current data value, $u(k+1)$.

$$\bar{u}(k+1) = (1-\phi)\ \bar{u}(k) + \phi\ u(k+1) \tag{116}$$

where ϕ is called the *forgetting factor* and $0 < \phi < 1$.

Working out the sequence of running averages from $u(0) = 0$, and $\bar{u}(0) = 0$:

$$\begin{aligned}
\bar{u}(1) &= (1-\phi)\ \bar{u}(0) + \phi\ u(1) = \phi\ u(1) \\
\bar{u}(2) &= (1-\phi)\ \bar{u}(1) + \phi\ u(2) = (1-\phi)\phi\ u(1) + \phi\ u(2) \\
\bar{u}(3) &= (1-\phi)\ \bar{u}(2) + \phi\ u(3) = (1-\phi)^2\phi\ u(1) + (1-\phi)\phi\ u(2) + \phi\ u(3) \\
&\ \vdots \\
\bar{u}(k) &= (1-\phi)^{k-1}\phi\ u(1) + (1-\phi)^{k-2}\phi\ u(2) + \cdots + (1-\phi)\phi\ u(k-1) + \phi\ u(k) \\
\bar{u}(k) &= \sum_{j=0}^{k-1} Y(j)\ u(k-j) \qquad \text{where} \qquad Y(j) = (1-\phi)^j\phi \quad \forall \quad j \geq 0
\end{aligned} \tag{117}$$

The most recent data point $u(k)$ is wighted by $Y(0)$ and the oldest data point $u(1)$ is weighted by $Y(k-1)$. Since $j \geq 0$ and $0 < \phi < 1$:

- $0 < Y(j) \leq \phi \ \forall\ j \geq 0$;
- the weights $Y(j)$ decrease exponentially with j;
- the largest weight, $Y(0) = \phi$, is on the most recent data;
- the smallest weight, $Y(k)$, is on the oldest data; and
- as k increases, the running average approaches the true weighted average

$$\lim_{k\to\infty} \sum_{j=0}^{k} Y(j) = 1$$

as shown in Figures 4 and 5.

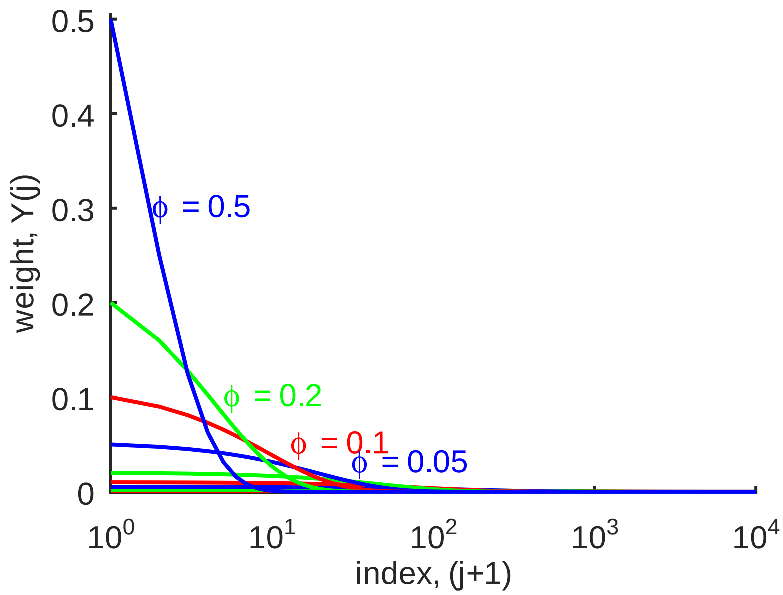

Figure 4. Weights of a running average for various values of the forgetting factor ϕ.

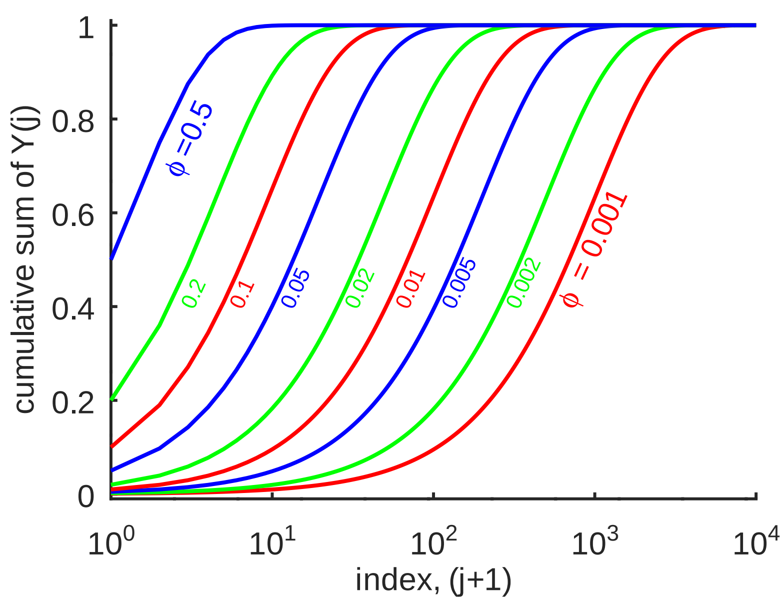

Figure 5. Cumulative sum of weights of a running average for various values of the forgetting factor ϕ.

Note that the running average at the k-th time step, $\bar{u}(k)$, involves the entire sequence of inputs, from $u(1)$ and up to and including $u(k)$, but older data contributes less and less to the running average. As a rule of thumb, the most recent $(5/\phi)$ points of data contribute significantly to the running average. For example, for $\phi = 0.01$, the most recent 500 points of data contribute significantly to the running average, and for $\phi = 0.1$, only the last 50 points contribute significantly to the running average.

Defining the state to be the running average, $x(k) = \bar{u}(k)$, the discrete time LTI system, found by inspection by by comparing the weights in (117), $Y(j) = (1-\phi)^j \phi$, to the Markov parameters in terms of state space matrices in (112), $Y(j) = CA^{j-1}B$, is

$$A = (1-\phi), \quad B = \phi, \quad C = (1-\phi), \quad D = Y(0) = \phi.$$

Figure 6 shows the running averages of a step input ($u(0) = 0$, $u(k) = 1 \, \forall \, k > 0$) and the running averages of sinusoids, $u(k) = \sin(k/5)+\sin(k/50)$ for $0 < k \leq 500$. Note that running averages with larger forgetting factors respond more rapidly and running averages with smaller forgetting factors are smoother.

32 Frequency Response in Discrete-Time

Consider the steady-state harmonic response of a discrete-time LTI system to harmonic inputs. The real-valued harmonic input, state, and output are represented as the sum of complex conjugates at a single frequency ω.

$$u(t) = u(\omega)e^{i\omega t} + u^*(\omega)e^{-i\omega t} \tag{118}$$
$$x(t) = x(\omega)e^{i\omega t} + x^*(\omega)e^{-i\omega t} \tag{119}$$
$$y(t) = y(\omega)e^{i\omega t} + y^*(\omega)e^{-i\omega t} \tag{120}$$

At discrete points in time $t_k = k\Delta t$, the states at t_{k+1} are

$$x(k+1) = x(\omega)e^{i\omega(k+1)\Delta t} + x^*(\omega)e^{-i\omega(k+1)\Delta t} \tag{121}$$

Substituting equations (118) - (121) into the discrete-time state-space equations, (92), and recognizing that the complex conjugate parts of the response ($x(\omega)e^{i\omega k\Delta t}$ and $x^*(\omega)e^{-i\omega k\Delta t}$) are linearly independent, we obtain

$$x(\omega)e^{i\omega(k+1)\Delta t} = Ax(\omega)e^{i\omega k\Delta t} + Bu(\omega)e^{i\omega k\Delta t} \tag{122}$$
$$y(\omega)e^{i\omega k\Delta t} = Cx(\omega)e^{i\omega k\Delta t} + Du(\omega)e^{i\omega k\Delta t} \tag{123}$$

Now, factoring-out the $e^{i\omega k\Delta t}$ from each term,

$$x(\omega)\ e^{i\omega\Delta t} = Ax(\omega) + Bu(\omega) \tag{124}$$
$$y(\omega) = Cx(\omega) + Du(\omega) \tag{125}$$

Solving for the outputs in terms of the inputs gives the frequency response function in terms of a state-space LTI model in the discrete-time domain,

$$y(\omega) = \left[\ C[e^{i\omega\Delta t}I - A]^{-1}B + D\ \right]\ u(\omega) \tag{126}$$
$$H(z) = C[zI - A]^{-1}B + D \tag{127}$$

where $z = e^{i\omega\Delta t}$. Equation (127) is analogous to the transfer function in terms of continuous-time state-space models, equation (63).

33 Laplace transform and z-transform

The Laplace transform of a continuous-time function $y(t)$ is given by

$$y(s) = \mathcal{L}[y(t)] = \int_0^\infty y(t)e^{-st}\ dt, \quad s \in \mathbb{C} \tag{128}$$

This one-sided Laplace transform applies to causal functions for which $y(t) = 0\ \forall\ t < 0$.

Now, if we sample $y(t)$ at discrete points in time spaced with interval Δt, $y(k\Delta t) = y(k) = y(t)\delta(t - k\Delta t)$, and take the Laplace transform of the sampled signal,

$$y(s) = \int_0^\infty \sum_{k=0}^{\infty} y(t)\delta(t - k\Delta t)\ e^{-st}\ dt\ . \tag{129}$$

Recall the property of the delta function,

$$\int_0^\infty f(t)\delta(t-\tau)\ dt = f(\tau) \quad (\forall\ \tau > 0), \tag{130}$$

so,

$$y(s) = \sum_{k=0}^{\infty} y(k\Delta t)e^{-sk\Delta t}\ dt = \sum_{k=0}^{\infty} y(k)\left(e^{s\Delta t}\right)^{-k}\ . \tag{131}$$

Defining $z = e^{s\Delta t}$, we arrive at the discrete-time version of the Laplace transform . . . the z-transform:

$$y(z) = \sum_{k=0}^{\infty} y(k)z^{-k} = \mathcal{Z}[y(k)]. \tag{132}$$

Now, we can apply the z-transform to the discrete-time convolution

$$y(z) = \mathcal{Z}\left[\sum_{i=0}^{k} Y(i)u(k-i)\right] = \sum_{k=0}^{\infty} z^{-k} \sum_{i=0}^{k} Y(i)u(k-i) \tag{133}$$

If $u(k) = 0\ \forall\ k < 0$, then $\sum_{i=0}^{k} Y(i)u(k-i) = \sum_{i=0}^{\infty} Y(i)u(k-i)$, so

$$\begin{aligned} y(z) &= \sum_{k=0}^{\infty} z^{-k} \sum_{i=0}^{\infty} Y(i)u(k-i) & (134)\\ &= \left[\sum_{i=0}^{\infty} Y(i)z^{-i}\right]\left[\sum_{k=0}^{\infty} z^{-(k-i)}u(k-i)\right] & (135)\\ &= H(z)\ u(z) & (136) \end{aligned}$$

So, convolution in the discrete-time domain is equivalent to multiplication in the z-domain, and the frequency-response is related to the Markov parameters via

$$H(z) = \sum_{i=0}^{\infty} Y(i) z^{-i} \tag{137}$$

The discrete-time state-space equations (92), the sequence of Markov parameters (112), and the frequency response function (127) are equivalent descriptions for discrete-time linear time-invariant systems.

Now consider the steady-state response to a sinusoidal input sequence

$$\begin{aligned}
u(k) &= \cos(\omega k \Delta t) = \frac{1}{2}\left(e^{\mathrm{i}\omega k\Delta t} + e^{-\mathrm{i}\omega k \Delta t}\right) \\
y(k) &= \sum_{i=0}^{\infty} Y(i) u(k-i) \\
&= \sum_{i=0}^{\infty} Y(i) \cos(\omega(k-i)\Delta t) \\
&= \sum_{i=0}^{\infty} Y(i) \frac{1}{2}\left(e^{\mathrm{i}\omega(k-i)\Delta t} + e^{-\mathrm{i}\omega(k-i)\Delta t}\right) \\
&= \frac{1}{2} e^{\mathrm{i}\omega k \Delta t} \sum_{i=0}^{\infty} Y(i) z^{-i} + \frac{1}{2} e^{-\mathrm{i}\omega k\Delta t} \sum_{i=0}^{\infty} Y(i) z^{i} \\
&= \frac{1}{2} e^{\mathrm{i}\omega k\Delta t} H(z) + \frac{1}{2} e^{-\mathrm{i}\omega k \Delta t} H^*(z) \\
&= |H(z)| \cos(\omega k \Delta t + \phi)
\end{aligned}$$

where the frequency-dependent phase angle of the sinusoidal response is given by

$$\tan\phi(z) = \frac{\mathsf{Im}(H(z))}{\mathsf{Re}(H(z))}$$

A graph of $|H(\omega)|$ and $\angle H(\omega)$ is called a *Bode plot*.

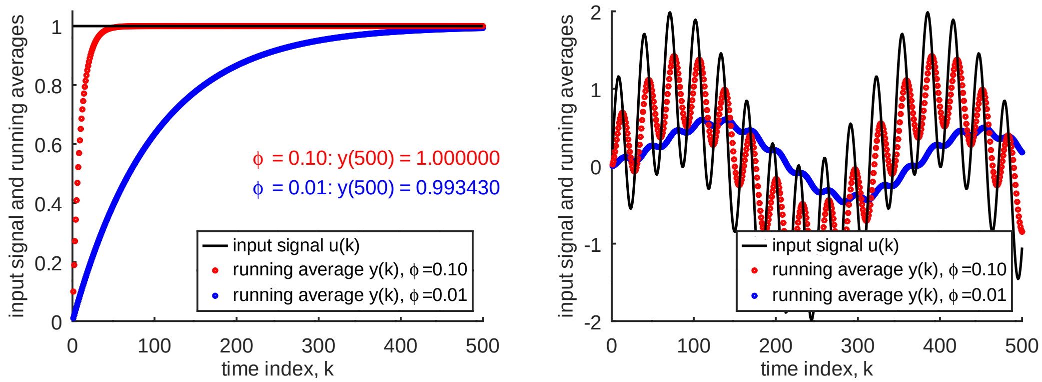

Figure 6. Time responses of the running average $\bar{u}(k)$ defined in system (116) to a step input and to a sinusoidal input. With $u(0) = 0$ and $u(1)$ to $u(500) = 1$, the arithmetic average is $500/501 \approx 0.998$. With $\phi = 0.10$, the running average at $k = 500$ has effectively *forgotten* that $u(0) = 0$. With $\phi = 0.01$, the running average at $k = 500$ clearly *remembers* that $u(0) = 0$. The forgetting factor ϕ and the frequency of sinusoidal components affect the time lag and the attenuation of those components, as shown in the right figure above and in Figure 7 below.

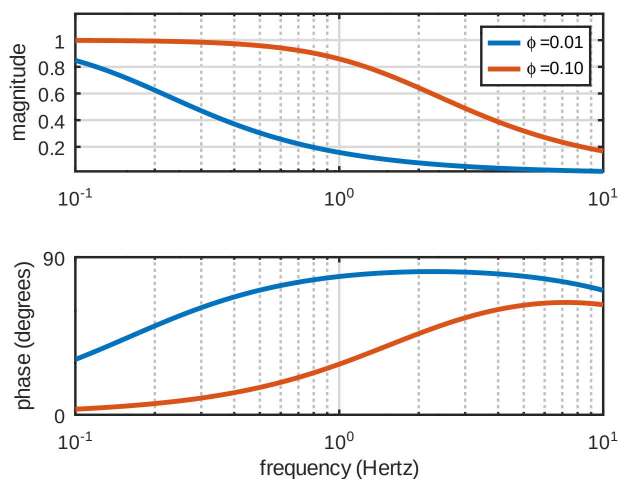

Figure 7. The Bode plot of the running average system (116) (assuming $\Delta t = 0.01$s) shows how the frequencies of sinusoidal components of an input time series affect the attenuation and the phase lag of corresponding components of the output time series. With $\Delta t = 0.01$s, the frequencies of the components shown in Figure 6 are 0.32 Hz and 3.2 Hz. Compare the magnitude $|\bar{u}(f)|$ and phase $\theta(f)$ shown above to the amplitude of $\bar{u}(t)$ and the time delay $\tau(f) = \theta(f)/(2\pi f)$ shown in Figure 6.

34 Liapunov Equations for Discrete-Time Systems

In discrete-time, the free response is

$$x(k+1) = Ax(k), \qquad x(0) = x_o \neq 0 \tag{138}$$

Defining a Liapunov function as the energy in the system at time k as,

$$V(k) = x^{\mathsf{T}}(k)Px(k), \qquad P > 0 \tag{139}$$

If the energy in free response decays monotonically, that is, if $V(k+1) - V(k) < 0 \ \forall \ x(k)$ and then the system is asymptotically stable.

$$\begin{aligned} V(k+1) - V(k) &= x^{\mathsf{T}}(k+1)Px(k+1) - x^{\mathsf{T}}(k)Px(k) \\ &= x^{\mathsf{T}}(k)A^{\mathsf{T}}PAx(k) - x^{\mathsf{T}}(k)Px(k) \\ &= x^{\mathsf{T}}(k)\,[A^{\mathsf{T}}PA - P]\,x(k) \end{aligned} \tag{140}$$

So, $[A^{\mathsf{T}}PA - P] < 0 \quad \Leftrightarrow \quad V(k+1) - V(k) < 0 \ \forall \ x(k)$, and the system $x(k+1) = Ax(k)$ is asymptotically stable (AS) if and only if there exist positive definite matrices P and Z that solve the left discrete-time Liapunov equation,

$$A^{\mathsf{T}}PA - P + Z = 0 \tag{141}$$

The following statements are equivalent:

- A is asymptotically stable
- all eigenvalues λ_i of A have magnitudes less than 1
- $\exists Z > 0$ s.t. $P > 0$ is a solution to $A^{\mathsf{T}}PA - P + Z = 0$
- the series $\sum_{k=0}^{\infty} A^{k\mathsf{T}}ZA^k$ converges .

The series $P = \sum_{k=0}^{\infty} A^{k\mathsf{T}}ZA^k$ solves $A^{\mathsf{T}}PA - P + Z = 0$.

$$\begin{aligned} P &= Z + A^{\mathsf{T}}PA \\ \sum_{k=0}^{\infty} A^{k\mathsf{T}}ZA^k &= Z + A^{\mathsf{T}} \sum_{k=0}^{\infty} A^{k\mathsf{T}}ZA^kA \\ &= Z + A^{\mathsf{T}} \sum_{k=1}^{\infty} A^{\mathsf{T}(k-1)}ZA^{k-1}A \\ Z + \sum_{k=1}^{\infty} A^{k\mathsf{T}}ZA^k &= Z + \sum_{k=1}^{\infty} A^{k\mathsf{T}}ZA^k \qquad \text{(since } A^0 = I\text{).} \end{aligned}$$

35 Controllability of Discrete-Time Systems

A discrete-time system with n states, r inputs, and dynamics matrix A is said to be *controllable* by inputs coming through input matrix B if there exists a sequence of inputs, $[u(0), u(1), \ldots, u(n-2), u(n-1), u(n)]$ that can bring the equilibrium state $x(0) = 0$ to any arbitrary state $x(n)$ within n steps.

Let's consider the sequence of states arising from an input sequence $u(k)$, $\ldots (k = 0, \ldots, n-1)$ starting from an initial state $x(0) = 0$.

$$
\begin{aligned}
x(1) &= Bu(0) \\
x(2) &= Ax(1) + Bu(1) = ABu(0) + Bu(1) \\
x(3) &= Ax(2) + Bu(2) = A^2Bu(0) + ABu(1) + Bu(2) \\
\vdots & \quad \vdots \\
x(n) &= A^{(n-1)}Bu(0) + \cdots + A^2Bu(n-3) + ABu(n-2) + Bu(n-1)
\end{aligned}
$$

$$
x(n) = \left[B \;,\; AB \;,\; A^2B \;,\; \cdots \;,\; A^{(n-2)}B \;,\; A^{(n-1)}B\right] \begin{bmatrix} u(n-1) \\ u(n-2) \\ u(n-3) \\ \vdots \\ u(1) \\ u(0) \end{bmatrix} \tag{142}
$$

Think of this last equation as a simple matrix-vector multiplication.

$$
x(n) = \mathcal{C}_n u_n \tag{143}
$$

The matrix $\mathcal{C}_n$ has n rows and nr columns and is called the controllability matrix. The column-vector u_n is the sequence of inputs, all stacked up on top of each other, into one long vector.

If the n rows of the controllability matrix $\mathcal{C}_n$ are linearly-independent, then any final state $x(n)$ can be reached through an appropriate selection of the control input sequence, u_n. If the rank of $\mathcal{C}_n$ equals n then there is at least one sequence of inputs u_n that can take the state from $x(0) = 0$ to any state $x(n)$ within n steps. So, if $\text{rank}(\mathcal{C}_n) = n$ then the pair (A, B) is controllable.

If $r > 1$ then the system is under-determined and the input sequence u_n to arrive at state $x(n)$ (in n steps), is computed using the *right* Moore-Penrose pseudo inverse, which minimizes $||u_n||_2^2$.

The covariance of an infinitely long sequence of state responses to independent impulses ($u(0) = I_r$) is the controllability gramian,

$$\mathcal{C}_\infty \mathcal{C}_\infty^\mathsf{T} = Q = \sum_{k=0}^{\infty} A^k B B^\mathsf{T} A^{k\mathsf{T}} \tag{144}$$

The element $Q_{(i,j)}$ is the covariance of the the inner product of the i-th state unit impulse responses with the j-th state unit impulse response, with unit impulses at each input. In other words, if $x_{iq}(k)$ is the response of the i-th state to a unit impulse at input q, then

$$Q_{(i,j)} = \sum_{k=0}^{\infty} \sum_{q=1}^{r} x_{iq}(k) x_{jq}(k). \tag{145}$$

The controllability gramian solves the right Liapunov equation

$$0 = AQA^\mathsf{T} - Q + BB^\mathsf{T} \tag{146}$$

These expressions are analogous to equations (74) and (75) in continuous-time. The following statements are equivalent:

- The pair (A, B) is controllable.
- The rank of $\mathcal{C}_\infty$ is n.
- The rank of Q is n.
- A matrix $Q > 0$ solves the right Liapunov equation $0 = AQA^\mathsf{T} - Q + BB^\mathsf{T}$

The singular-value decomposition of the controllability matrix equation

$$x(n) = \sum_i \sigma_{\mathcal{C}i} \, u_{\mathcal{C}i} \, (v_{\mathcal{C}i}^\mathsf{T} \, u_n) \tag{147}$$

shows that the sequence of inputs proportional to $v_{\mathcal{C}1}$ couples most strongly to the states. Input sequences that lie entirely in the kernel of $\mathcal{C}_n$ have no effect on the state. Likewise, state vectors proportional to the eigenvector of Q with the largest eigenvalue are most strongly affected by inputs u_n And state vectors proportional to an eigenvector of Q with an eigenvalue of zero (if there is one) can not be attained by the control input $Bu(k)$.

36 Observability of Discrete-Time Systems

A system with n states, m outputs, and dynamics matrix A is said to be *observable* by outputs coming out through output matrix C if there exists a sequence of outputs $[y(0),\ y(1),\ \ldots,\ y(n-3),\ y(n-2),\ y(n-1)]$ from which the initial state $x(0)$ can be uniquely determined.

Let's consider the sequence of outputs arising from a sequence of states, in free response from some initial condition $x(0)$. (The initial condition is not equal to zero.)

$$
\begin{aligned}
y(0) &= Cx(0) \\
y(1) &= Cx(1) = CAx(0) \\
y(2) &= Cx(2) = CAx(1) = CA^2x(0) \\
\vdots \quad & \quad \vdots \\
y(n-1) &= Cx(n-1) = CA^{(n-1)}x(0)
\end{aligned}
$$

$$
\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{(n-1)} \end{bmatrix} x(0) \tag{148}
$$

Think of this last equation as a simple matrix-vector multiplication.

$$
y_n = \mathcal{O}_n x(0) \tag{149}
$$

The matrix $\mathcal{O}_n$ has nm rows and n columns and is called the observability matrix. The column-vector y_n is the sequence of outputs, all stacked up on top of each other, into a long vector.

If the n columns of the observability matrix, $\mathcal{O}_n$, are linearly-independent, then any initial state $x(0)$ can be determined from the associated sequence of n free-responses. If $\mathcal{O}_n$ has rank n then the set of all vectors y_n can be transformed into a set of vectors x_0 that fill an n-dimensional vector space, via the matrix inverse (or pseudo-inverse) of $\mathcal{O}_n$.

If $m > 1$ then the matrix equation is over determined, and the initial state $x(0)$ associated with the free response outputs y_n is computed with the *left* Moore-Penrose pseudo inverse, which minimizes $||y_n - \mathcal{O}_n x(0)||_2^2$.

The covariance of an infinitely long sequence of output responses to n independent unity initial conditions ($x(0) = I_n$) is the observability gramian,

$$\mathcal{O}_\infty^\mathsf{T} \mathcal{O}_\infty = P = \sum_{k=0}^{\infty} A^{k\mathsf{T}} C^\mathsf{T} C A^k \tag{150}$$

The element $P(i,j)$ is the covariance of the inner product of the two free output responses one evolving from $x_i(0) = 1$ and the other from $x_j(0) = 1$ (with $x_k(0) = 0, \forall k \neq i, j$). In other words, if $y_{pi}(k)$ is the free response of the p-th output to the initial state $x_i(0) = 1$, $x_j(0) = 0, \forall\ i \neq j$, then

$$P_{(i,j)} = \sum_{k=0}^{\infty} \sum_{p=1}^{m} y_{pi}(k) y_{pj}(k). \tag{151}$$

The observability gramian solves the left Liapunov equation

$$0 = A^\mathsf{T} P A - P + C^\mathsf{T} C \tag{152}$$

These expressions are analogous to equations (72) and (73) in continuous-time. The following statements are equivalent:

- The pair (A, C) is observable.
- The rank of $\mathcal{O}_\infty$ is n.
- The rank of P is n.
- A matrix $P > 0$ solves the left Liapunov equation
  $0 = A^\mathsf{T} P A - P + C^\mathsf{T} C$

The singular-value decomposition of the observability matrix equation

$$y(n) = \sum_i \sigma_{\mathcal{O}i}\ u_{\mathcal{O}i}\ (v_{\mathcal{O}i}^\mathsf{T}\ x(0)) \tag{153}$$

shows that an initial state $x(0)$ proportional to $v_{\mathcal{O}1}$ couples most strongly to the outputs. Initial states that lie entirely in the kernel of $\mathcal{O}_n$ (if it exists) have no output through $y = Cx$. Likewise, initial states proportional to the eigenvector of P with the largest associated eigenvalue contribute most to the output covariance. And initial states proportional to an eigenvector of P with an eigenvalue of zero (if there are any) do not affect the output covariance.

37 H_2 norms of discrete-time LTI systems

The H_2 norm of asymptotically-stable discrete-time models with $D = 0$ is expressed in terms of the controllability gramian and observability gramian satisfying the right and left discrete-time Liapunov equations (146) and (152) and equations (77) and (80).

Because the frequency content of signals in discrete-time is limited to the Nyquist interval, the H_2 norm in terms of the frequency response of discrete-time systems involves integration around the unit circle, $||z|| = 1$

$$||H||_2^2 = \frac{1}{2\pi} \int_{-\pi}^{+\pi} \mathrm{tr}\left[H(e^{\mathrm{i}\omega\Delta t})^{\mathsf{T}} H(e^{\mathrm{i}\omega\Delta t}) \right] \, d\omega \tag{154}$$

where $H(e^{\mathrm{i}\omega\Delta t}) = H(z)$ as defined for discrete-time systems in equation (127).

In terms of unit impulse responses (Markov parameters), the H_2 norm of a discrete time system is

$$||H||_2^2 = \sum_{k=0}^{\infty} ||Y(k)||_{\mathrm{F}}^2 \tag{155}$$

where $Y(0) = D$ and $Y(k) = CA^{k-1}B$ for $k > 0$.

Note that in continuous-time systems, $\lim_{\omega\to\infty}[H(\omega)] = D$. Referring to the frequency-domain interpretation of the H_2 norm, equation (78), we see that for systems with $D \neq 0$, the integral of $||H(\omega)||_{\mathrm{F}}$ over $-\infty < \omega < \infty$ is not finite, and so the H_2 norm can not be defined in this case. In the time domain, the unit impulse $u_i(t) = I_r\delta(t)$ has responses $H(t) = Ce^{At}B + D\delta(t)$. In this case the integral of $||H(t)||_{\mathrm{F}}$ over $0 < t < \infty$ involves the integral of $\delta^2(t)$, which is not finite, and so the H_2 norm can not be defined in this case either. The facts that H_2 norms of exactly proper continuous-time systems are not finite, and that they are for discrete-time systems is a consequence of the facts that Dirac delta is defined only in terms of convolution integrals, and that $u(0) = 1$ in discrete time is a perfectly reasonable statement. A low-pass filtered Dirac-delta is a sinc function, which is square integrable. So, a norm-equivalency of discrete-time systems derived from continuous-time systems necessarily implies that continuous-time inputs have no power outside of the Nyquist interval.

38 .m-functions

abcddim.m http://www.duke.edu/~hpgavin/abcddim.m

```
function [n, r, m] = abcddim (A, B, C, D)
% Usage: [n, r, m] = abcddim (A, B, C, D)
%
% Check for compatibility of the dimensions of the matrices defining
% the linear system (A, B, C, D).
%
% Returns n = number of system states,
%         r = number of system inputs,
%         m = number of system outputs.
%
% Returns n = r = m = -1 if the system is not compatible.

% A.S. Hodel <scotte@eng.auburn.edu>

  if (nargin ~= 4)
    error ('usage: abcddim (A, B, C, D)');
  end

  n = -1; r = -1; m = -1;

  [an, am] = size(A);
  if (an ~= am), error ('abcddim: A is not square'); end

  [bn, br] = size(B);
  if (bn ~= an)
     error (sprintf('abcddim: A and B are not compatible, A:(%dx%d) B:(%dx%d)',am,an,bn,br))
  end

  [cm, cn] = size(C);
  if (cn ~= an)
     error (sprintf('abcddim: A and C are not compatible, A:(%dx%d) C:(%dx%d)',am,an,cm,cn))
  end

  [dm, dr] = size(D);
  if (cm ~= dm)
     error (sprintf('abcddim: C and D are not compatible, C:(%dx%d) D:(%dx%d)',cm,cn,dm,dr))
  end
  if (br ~= dr)
     error (sprintf('abcddim: B and D are not compatible, B:(%dx%d) D:(%dx%d)',bn,br,dm,dr))
  end

  n = an;
  r = br;
  m = cm;

% ------------------------------------------------------------------ abcddim.m
```

lsym.m http://www.duke.edu/~hpgavin/lsym.m

```
function y = lsym(A,B,C,D,u,t,x0, ntrp)
% y = lsym ( A, B, C, D, u, t, x0, ntrp )
% transient response of a continuous-time linear system to arbitrary inputs.
%                     dx/dt = Ax + Bu
%                       y   = Cx + Du
%
%       A    :     dynamics     matrix                          (n by n)
%       B    :     input        matrix                          (n by r)
%       C    :     output       matrix                          (m by n)
%       D    :     feedthrough  matrix                          (m by r)
%       u    :     matrix of sampled inputs                     (r by p)
%       t    :     vector of uniformly spaced points in time    (1 by p)
%       x0   :     vector of states at the first point in time  (n by 1)
%       ntrp :     'zoh' zero order hold, 'foh' first order hold (default)
%       y    :     matrix of the system outputs                 (m by p)

 if (nargin < 8) , ntrp = 'foh'; end

 [n,r,m] = abcddim(A,B,C,D);        %  matrix dimensions and compatability check

 points = size(u,2);                % number of data points

 dt =  t(2) - t(1);                 % uniform time-step value

 % continuous-time to discrte-time conversion ...
 if strcmp(lower(ntrp),'zoh')       % zero-order hold on inputs
   M    = [ A B ; zeros(r,n+r) ];
 else                               % first-order hold on inputs
   M    = [ A B zeros(n,r) ; zeros(r,n+r) eye(r) ; zeros(r,n+2*r) ];
 end
 eMdt  = expm(M*dt);                % matrix exponential
 Ad    = eMdt(1:n,1:n);             % discrete-time dynamics matrix
 Bd    = eMdt(1:n,n+1:n+r);         % discrete-time input matrix
 if strcmp(lower(ntrp),'zoh')
   Bd0   = Bd;
   Bd1   = zeros(n,r);
 else
   Bd_   = eMdt(1:n,n+r+1:n+2*r);   % discrete-time input matrix
   Bd0   = Bd   -   Bd_ / dt;       % discrete-time input matrix for time p
   Bd1   = Bd_ / dt;                % discrete-time input matrix for time p+1
 end

% B and D for discrete time system
% Bd_bar = Bd0 + Ad*Bd1;
% D_bar  = D    + C*Bd1;

% Markov parameters for the discrete time system with ZOH
% Y0 = D_bar
% Y1 = C * Bd_bar
% Y2 = C * Ad * Bd_bar
% Y3 = C * Ad^2 * Bd_bar

 % initial conditions are zero unless specified

 if ( nargin < 7 )
   x0 = zeros(n,1);                 % initial conditions are zero
 end

 y = zeros(m,points);               % memory allocation for the output
```

dlsym.m http://www.duke.edu/~hpgavin/dlsym.m

```
function y = dlsym(A,B,C,D,u,t,x0)
% y = dlsym(A,B,C,D,u,t,x0)
% simulates the response of a discrete-time linear system to arbitrary inputs
%
%                x(k+1) = A x(k)  +  B u(k)
%                y(k)   = C x(k)  +  D u(k)
%
%       A    :   n by n  dynamics     matrix
%       B    :   n by m  input        matrix
%       C    :   l by n  output       matrix
%       D    :   l by m  feedthrough  matrix
%       u    :   l by p  matrix of sampled inputs
%       t    :   1 by p  vector of uniformly spaced points in time
%       x0   :   n by 1  vector of initial states, defaults to zero
%       y    :   m by p  matrix of the system outputs
%

 [n,m,l] = abcddim(A,B,C,D);

 points = size(u,2);              % number of data points

 if ( nargin < 6 )
   t = [1:points];
 end

 if ( nargin == 7 )
   x = x0;                        % initial conditions for the state
 else
    x = zeros(n,1);               % initial conditions are zero
 end

 y = NaN(l,points);             % memory allocation for the output

 y(:,1) = C * x  +  D * u(:,1);

 for p = 2:points

     x = A * x  +  B * u(:,p-1);

     y(:,p) = C * x  +  D * u(:,p);

 end

end % ------------------------------------------------------------ dlsym.m
% 2021-07-19 ...
%   replaced ...   x = A * x  +  B * u(:,p);
%   ... with ...   x = A * x  +  B * u(:,p-1);
```

damp.m http://www.duke.edu/~hpgavin/damp.m

```
function [wn,z] = damp(a,delta_t)
% [wn,z] = damp(A,delta_t)
% DAMP  Natural frequency and damping factor for continuous or discrete systems.
%
% damp(A) displays a table of the natural frequencies and
% damping ratios for the continuous-time dynamics matrix.
%
% damp(A,delta_t) displays a table of the natural frequencies
% damping ratios for the discrete-time dynamics matrix.
% with a sample time step of delta_t
%
% [wn,z] = damp(A) or [wn,z] = damp(A,delta_t) returns the vectors
%  wn and z of the natural frequencies and damping ratios, without
%  displaying the table of values.
%
%  The variable A can be in one of several formats:
%
% (1) If A is square, it is assumed to be the state-space dynamics matrix.
%
% (2) If A is a row vector, it is assumed to be a vector of the
%     polynomial coefficients from a transfer function.
%
% (3) If A is a column vector, it is assumed to contain root locations.
%

[m,n] = size(a);

if (n<1 || m<1), wn=0; z=0; return; end

if (m == n)
  r = eig(a);
elseif (m == 1)
  r = (roots(a));
elseif (n == 1)
  r = a;
else
  error('The variable A must be a vector or a square matrix.');
end

if ( nargin == 2 ), r = log(r)/delta_t; end     % discrete time system

for k = 1:n
  wn(k)   = abs(r(k));
  z(k)    = - (real(r(k)) - 2*eps) / (wn(k) + 2*eps);
end

[wns,idx] = sort(abs(wn));      % sort by increasing natural frequency
wn = wn(idx);
z  = z(idx);
r  = r(idx);
wd = wn .* sqrt( abs ( z.^2 - 1 ) );

if nargout == 0                 % Display results on the screen.
  fprintf(' \n');
  fprintf('     Natural                Damped  \n');
  fprintf('    Frequency              Frequency      Eigenvalue \n');
  fprintf('    (cyc/sec)    Damping  (cyc/sec)    real         imag \n');
  fprintf('    -----------------------------------------------------\n');

  for idx = 1:1:n
```

```
    fprintf('    %10.5f  %10.5f    %10.5f   %10.5f  %10.5f  \n', wn(idx)/(2*pi) , z(idx), wd(idx)/(2*pi),
  end

  return % Suppress output
end

% ------------------------------------------------------------------------------------ DAMP
```

mimoBode.m http://www.duke.edu/~hpgavin/mimoBode.m

```
function [mag,pha,G] = mimoBode(A,B,C,D,w,dt,figno,ax,leg,tol)
% [mag,pha,G] = mimoBode(A,B,C,D,w,dt,figno,ax,leg,tol)
% plots the magnitude and phase of the
% steady-state harmonic reponse of a MIMO linear dynamic system.
% where:
%  A,B,C,D are the standard state-space matrices of a dynamic system
%  w is a vector of frequencies          (default: w = logspace(-2,2,200)*2*pi
%  dt is the sample period               (default: dt = [] for continuous time)
%  ax is either x,y,n, or b to indidate wich ax should be log-scaled ...
%   x, y, neither, or both.  The default is both.  The y-axis for the phase plot
%  is always linearly-scaled.
%  mag and pha are the magnitude and phase of the frequency response fctn matrix
%  mag and pha have dimension ( length(w) x m x r )

%  Henri Gavin, Dept. Civil Engineering, Duke University, henri.gavin@duke.edu

% Krajnik, Eduard, 'A simple and reliable phase unwrapping algorithm,'
% http://www.mathnet.or.kr/mathnet/paper_file/Czech/Eduard/phase.ps

 if (nargin < 10) tol = 1e-26;  end             % default rcond level
 if (nargin <  9) leg = []; end
 if (nargin <  8) ax = 'n';     end             % default plot formatting
 if (nargin <  7) figno = 100;  end             % default plot formatting
 if (nargin <  6) dt = [];      end             % default to continuous time
 if (nargin <  5) w = logspace(-2,2,200)*2*pi;  end     % default frequency axis

 [n,r,m] = abcddim(A,B,C,D);              % check for compatibile dimensions
 nw = length(w);

 lw = 3;                                  % line width

%warning off

 In = eye(n);

% continuous time or discrete time ...
 if (length(dt) == 0) sz = 1i*w; else sz = exp(1i*w*dt); end

 G   = zeros(nw,m,r); % allocate memory for the frequency response function, g
 mag = NaN(nw,m,r);
 pha = NaN(nw,m,r);

 for ii=1:nw  % compute the frequency response function, G
   G(ii,:,:) = C *  ((sz(ii)*eye(n)-A) \  B)  + D; % (sI-A) is ill-conditioned
%  [u,s,v] = svd( sz(ii)*eye(n) - A );                  % SVD of (sI-A)
%  idx = max( find( diag(s) > s(1,1)*tol ) );
%  char_eq_inv = v(:,1:idx) * inv(s(1:idx,1:idx)) * u(:,1:idx)';
%  G(ii,:,:) = C * char_eq_inv * B + D;
 end
```

```

 mag = abs(G);
 pha(1,:,:)    = -atan2(imag(G(1,:,:)),real(G(1,:,:))); % -ve sign!
 pha1 = -atan2(imag(G(1,:,:)),real(G(1,:,:)));  % -ve sign!
%pha1 = reshape( ones(nw-1,1,1)*pha1 , [nw-1,m,r] );
 pha1 = repmat( pha1, [nw-1,1,1] );
 pha(2:nw,:,:) = pha1 - cumtrapz(angle(G(2:nw,:,:)./G(1:nw-1,:,:)));

 if length(dt) == 1          % remove out-of-Nyquest range values in DT systems
   w_out = find(w>pi/dt);   % frequencies outside of the Nyquist range
   mag(w_out) = NaN;
   pha(w_out) = NaN;
 end

 if (figno > 0)  % PLOTS ===================================================

  figure(figno);
   clf
   for k=1:r
    subplot(2,r,k)
     if (ax == 'x')
       semilogx(w/2/pi, mag(:,:,k), 'LineWidth', lw); % plot the magnitude resp
     elseif (ax == 'y')
       semilogy(w/2/pi, mag(:,:,k), 'LineWidth', lw); % plot the magnitude resp
     elseif (ax == 'n')
       plot(w/2/pi, mag(:,:,k), 'LineWidth', lw );    % plot the magnitude resp
     else
       loglog(w/2/pi, mag(:,:,k), 'LineWidth',lw );   % plot the magnitude resp
     end
     if (nargin > 8), legend(leg); end

     axis([ min(w)/2/pi , max(w)/2/pi , min(min(min(mag))) , 1.2*max(max(max(mag))) ]);
     if k == 1, ylabel('magnitude'); end
     grid on

    subplot(2,r,k+r)
     if (ax == 'n' || ax == 'y')
      plot(w/(2*pi), pha(:,:,k)*180/pi, 'LineWidth', lw )      % plot the phase
     else
      semilogx(w/(2*pi), pha(:,:,k)*180/pi, 'LineWidth', lw ); % plot the phase
     end
%    if (nargin > 9), legend(leg); end
     pha_min = floor(min(min(min(pha*180/pi))/90))*90;
     pha_max =  ceil(max(max(max(pha*180/pi))/90))*90;
     set (gca, 'ytick', [pha_min : 90 : pha_max ])
     axis([ min(w)/2/pi max(w)/2/pi pha_min pha_max ]);
     xlabel('frequency (Hertz)')
     if k == 1, ylabel('phase (degrees)'); end
     grid on

   end
 end

% ----------------------------------------------------------------- MIMOBODE
```

dliap.m http://www.duke.edu/~hpgavin/dliap.m

```
function P = dliap(A,X)
% function P = dliap(A,X)
% Solves the Liapunov equation A'*P*A - P + X = 0 for P by transforming
% the A & X matrices to complex Schur form, computes the solution of
% the resulting triangular system, and transforms this solution back.
% A and X are square matrices.

% http://www.mathworks.com/matlabcentral/newsreader/view_thread/16018
% From: daly32569@my-deja.com
% Date: 12 Apr, 2000 23:02:27
% Downloaded: 2015-08-04

% Transform the matrix A to complex Schur form
% A = U * T * U' ... T is upper-triangular, U*U' = I
[U,T] = schur(complex(A)); % force complex schur form since A is often real

% Now ... P - (U*T'*U')*P*(U*T*U') = X   ... which means ...
% U'*P*U - (T'*U')*P*(U*T) = U'*X*U
% Let Q = U'*P*U yields, Q - T'*Q*T = U'*X*U = Y

% Solve for Q = U'*P*U by transforming X to Y = U'*X*U
% Therefore, solve: Q - T*Q*T' = Y ...  for Q
% Save memory by using "P" for Q.
dim = size(A,1);
Y = U' * X * U;
T1 = T;
T2 = T';
P  = Y; % Initialize P ... that is, initialize Q
for col = dim:-1:1,
  for row = dim:-1:1,
    P(row,col) = P(row,col) + T1(row,row+1:dim)*(P(row+1:dim,col+1:dim)*T2(col+1:dim,col));
    P(row,col) = P(row,col) + T1(row,row)*(P(row,col+1:dim)*T2(col+1:dim,col));
    P(row,col) = P(row,col) + T2(col,col)*(T1(row,row+1:dim)*P(row+1:dim,col));
    P(row,col) = P(row,col) / (1 - T1(row,row)*T2(col,col));
  end
end
% U*P*U' - U*T1*P*T1'*U' - X
% Convert Q to P by P = U'*Q*U.
P = U*P*U';
% A*P*A' - P + X    % check that this is zero, or close to it.
```

invzero.m http://www.duke.edu/~hpgavin/invzero.m

```
function zz = invzero(A,B,C,D,tol)
% zeros = invzero(A,B,C,D,tol)
% invariant and decoupling zeros of a continuous-time LTI system
% the condition for an invariant zero is that the pencil [ zI-A, -B; C D ]  is
% rank deficient.  For zeros that are not poles (i.e., for minimal realizaitons)
% invariant zeros, z, make H(z) rank-deficient.
% method: use QZ decomposition to find the general eigenvalues of the
% Rosenbrock system matrix,  padded with zero or randn to make it square.
% tol : tolerance value for rank determination,  default = 1e-4

 [nn,rr,mm] = abcddim(A,B,C,D);

 if nargin < 5, tol = 1e-6; end

% make the system square by padding with zeros or randn, as needed (cluge?)
  re = mm-rr;
  me = rr-mm;
  rm = max(mm,rr);

  zi = [];
  for iter = 1:4

    Bx = B;  Cx = C;  Dx = D;
    if iter == 1  % zero padding for decoupling zeros
      if mm > rr, Bx = [ B , zeros(nn,re) ]; Dx = [ D , zeros(mm,re) ]; end
      if mm < rr, Cx = [ C ; zeros(me,nn) ]; Dx = [ D ; zeros(me,rr) ]; end
    else           % randn padding for invariant zeros
      if mm > rr, Bx = [ B , randn(nn,re) ]; Dx = [ D , randn(mm,re) ]; end
      if mm < rr, Cx = [ C ; randn(me,nn) ]; Dx = [ D ; randn(me,rr) ]; end
    end

    abcd = [ -A , -Bx ; Cx , Dx ];       % Rosenbrock System Matrix
    ii = [ eye(nn) , zeros(nn,rm) ; zeros(rm,nn+rm) ];
    zz = -eig ( abcd , ii, 'qz' );
    zz = zz(isfinite(zz));
    if iter == 1
      z1 = zz;
    else
      zi = [ zi , zz ];
    end
  end

 zz = [ z1 ; intersecttol(zi, tol) ];

 idxR = find(abs(imag(zz)) < 1e-10 ); zz(idxR) = real(zz(idxR));  % real zeros
 idxC = find(abs(imag(zz)) > 1e-10 );                             % complex zeros

 zz = [ zz(idxR) ; zz(idxC) ; conj(zz(idxC)) ];      % complex conj pairs

 nz  = length(zz);

% Are both the pencil of the Rosenbrock System Matrix _and_
% the transfer function matrix rank deficient??
% confirm that all the zeros are invariant zeros

 nrcABCD = min(size([A B;C D]));
 min_mr = min(size(D));
 pp = eig(A);

 good_zero_index = [];
```

39 Numerical Example 1: a spring-mass-damper oscillator

For the single degree of freedom oscillator of Section 2, let's say $m = 2$ ton, $c = 1.4$ N/mm/s, $k = 6.8$ N/mm, $d_o = 5.5$ mm and $v_o = 2.1$ mm/s. Note that these units are consistent. (1 N)=(1 kg)(1 m/s^2)=(1 ton)(1 mm/s^2)

For these values, the linear time invariant system description of equations (9) and (10) become

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -3.4 & -0.7 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u(t) \ , \qquad x(0) = \begin{bmatrix} 5.5 \\ 2.1 \end{bmatrix} \tag{156}$$

$$y(t) = \begin{bmatrix} 6.8 & 1.4 \\ -3.4 & -0.7 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} u(t) \tag{157}$$

1. What are the natural frequencies and damping ratios of this system?

```
>> A = [ 0 1 ; -3.4 -0.7 ]     % the dynamics matrix

A =  0.00000   1.00000
    -3.40000  -0.70000

>> eig(A)                      % eigenvalues of the dynamics matrix

ans = -0.3500 + 1.8104i
      -0.3500 - 1.8104i

>> wn = abs(eig(A))            % absolute values of the eig's of A are omega_n

wn =  1.8439
      1.8439

>> z = -real(eig(A)) ./ wn     % ratio of real eig(A) to omega_n is damping ratio

z =  0.18981
     0.18981

>> wd = imag(eig(A))           % imaginary parts of the eig's of A are omega_d

wd =  1.8104
     -1.8104
```

So, the natural frequency is 1.84 rad/s (the same as $\sqrt{k/m}$), the damping ratios is 19% (the same as $c/\sqrt{4mk}$), and the damped natural frequency is 1.81 rad/s (the same as $\omega_{\rm n}\sqrt{|\zeta^2 - 1|}$).
These calculations can be done in one step with the m-file **damp.m**

```
>> damp(A)

         Natural                    Damped
        Frequency                  Frequency        Eigenvalue
        (cyc/sec)    Damping       (cyc/sec)     real         imag
        ----------------------------------------------------------
        0.29347     0.18981        0.28813     -0.35000      1.81039
        0.29347     0.18981        0.28813     -0.35000     -1.81039
```

2. What is the discrete-time system realization for $\Delta t = 0.01$ s?

```
>> dt = 0.01;                                   % time step, s
>> Ad = expm(A*dt);  Bd = A\(Ad-eye(2))*B;      % continuous-to-discrete-time

Ad =   0.9998304   0.0099645
      -0.0338794   0.9928552

Bd =   2.4941e-05
       4.9823e-03

>> damp(Ad,dt)            % check the dynamics of discrete-time system

         Natural                    Damped
        Frequency                  Frequency        Eigenvalue
        (cyc/sec)    Damping       (cyc/sec)     real         imag
        ----------------------------------------------------------
         0.29347     0.18981     0.28813       -0.35000      1.81039
         0.29347     0.18981     0.28813       -0.35000     -1.81039
```

The following analyses are carried out for the continuous-time system model and can also be equivalently carried out for the discrete-time system model.

3. What is the free response of this system to the specified initial conditions x_o?

```
>> dt = 0.01;                          % time step value, sec
>> n = 1000;                           % number of time steps
>> t = [0:n-1]*dt;                     % time values, starting at t=0
>> xo = [ 5.5 ; 2.1];                  % initial state (mm, mm/s)
>> C = [ 6.8 1.4 ; -3.4 -0.7 ];        % output matrix
>> y = zeros(2,n);                     % initialize outputs
>> for k=1:n
>      y(:,k) = C*expm(A*t(k))*xo;
>  end
>> plot(t,y)
>>  legend('foundation force, N', 'mass acceleration, mm/s^2')
>>  xlabel('time, s')
>>  ylabel('outputs, y_1 and y_2')
```

The free response is plotted in figure 8.

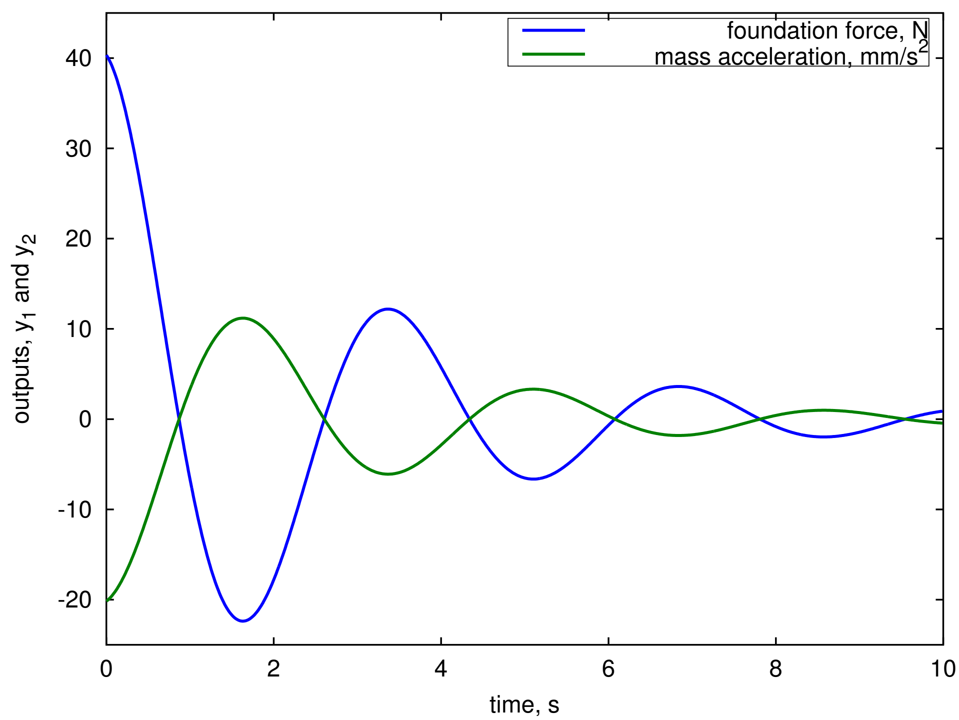

Figure 8. Free response of the linear time invariant system given in equations (156) and (157).

4. What is the response of this system to a unit impulse, $u(t) = \delta(t)$?

```
>> B = [ 0 ; 0.5 ];                  % input matrix
>> h = zeros(2,n);                   % initialize unit impulse responses
>> for k=1:n
>       h(:,k) = C*expm(A*t(k))*B;
>  end
>>  plot(t,h)
>>   legend('foundation force, N', 'mass acceleration, mm/s^2')
>>   xlabel('time, s')
>>   ylabel('unit impulse responses, h_1(t) and y_2(t)')
```

The unit impulse response is plotted in figure 9. Note that at $h(0) = CB$, which is not necessarily zero. .

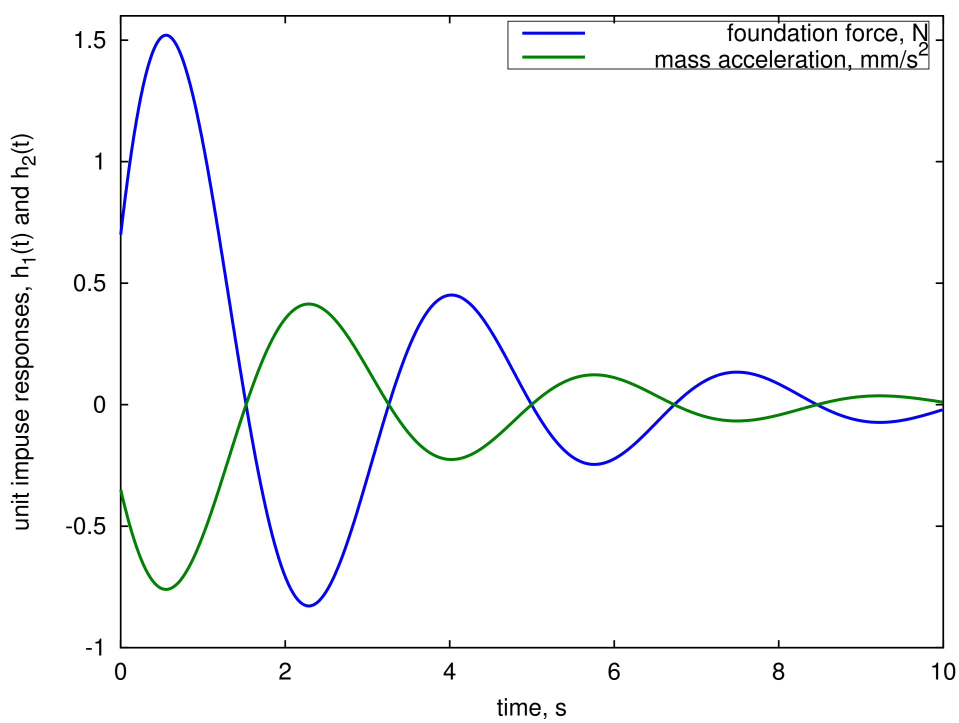

Figure 9. Unit impulse response of the linear time invariant system given in equations (156) and (157).

5. What is the response of this system to external forcing $u(t) = 50\cos(\pi t)$?

```
>> D = [ 0 ; 0.5 ];             % feedthrough matrix
>> u = 50 * cos(pi*t);          % external forcing
>> y = lsym(A,B,C,D,u,t,xo);    % use the "lsym" command
>>  plot(t,y)
>>   legend('foundation force, N', 'mass acceleration, mm/s^2')
>>   xlabel('time, s')
>>   ylabel('forced harmonic responses, y_1(t) and y_2(t)')
```

The forced harmonic response is plotted in figure 10.

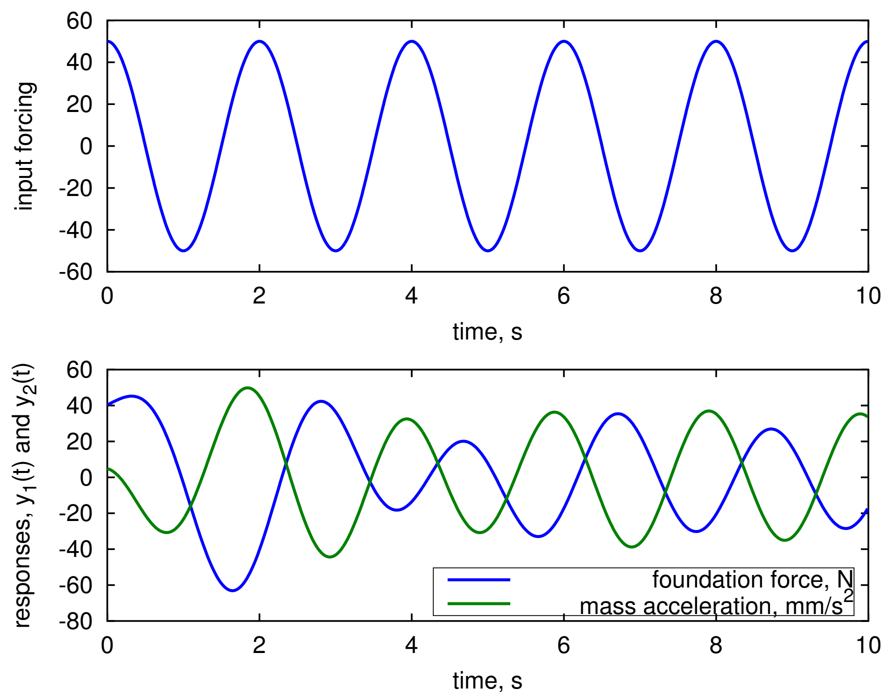

Figure 10. Forced response of the linear time invariant system given in equations (156) and (157).

6. What is the frequency response function from $u(t)$ to $y(t)$ for this system?

```
>> w = 2*pi*logspace(-1,0,100);          % frequency axis data
>> bode(A,B,C,D,1,w);
```

The magnitude and phase of the steady-state forced harmonic response is plotted in figure 11. Note how the magnitude and phase of the frequency response shown in figure 11 can be used to predict the steady state response of the system to a forcing of $u = 10\cos(\pi t)$. (The forcing frequency is 0.5 Hertz).

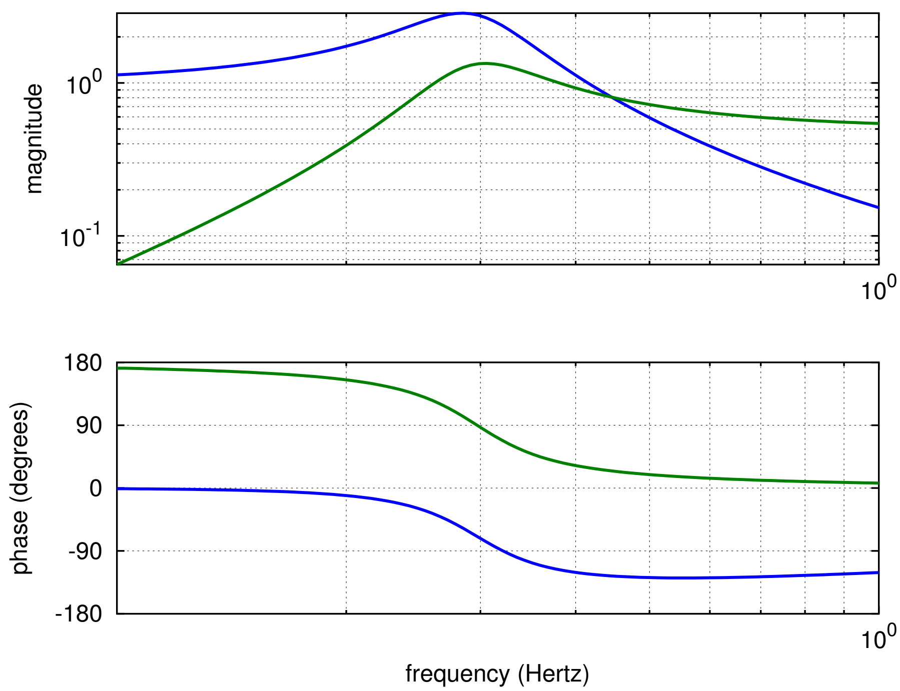

Figure 11. Frequency response of the linear time invariant system given in equations (156) and (157).

The Laplace-domain transfer functions of the system are plotted in figure 12.

This is a fairly simple example. Nevertheless, by simply changing the definitions of the system matrices, A, B, C, and D, and of the input forcing

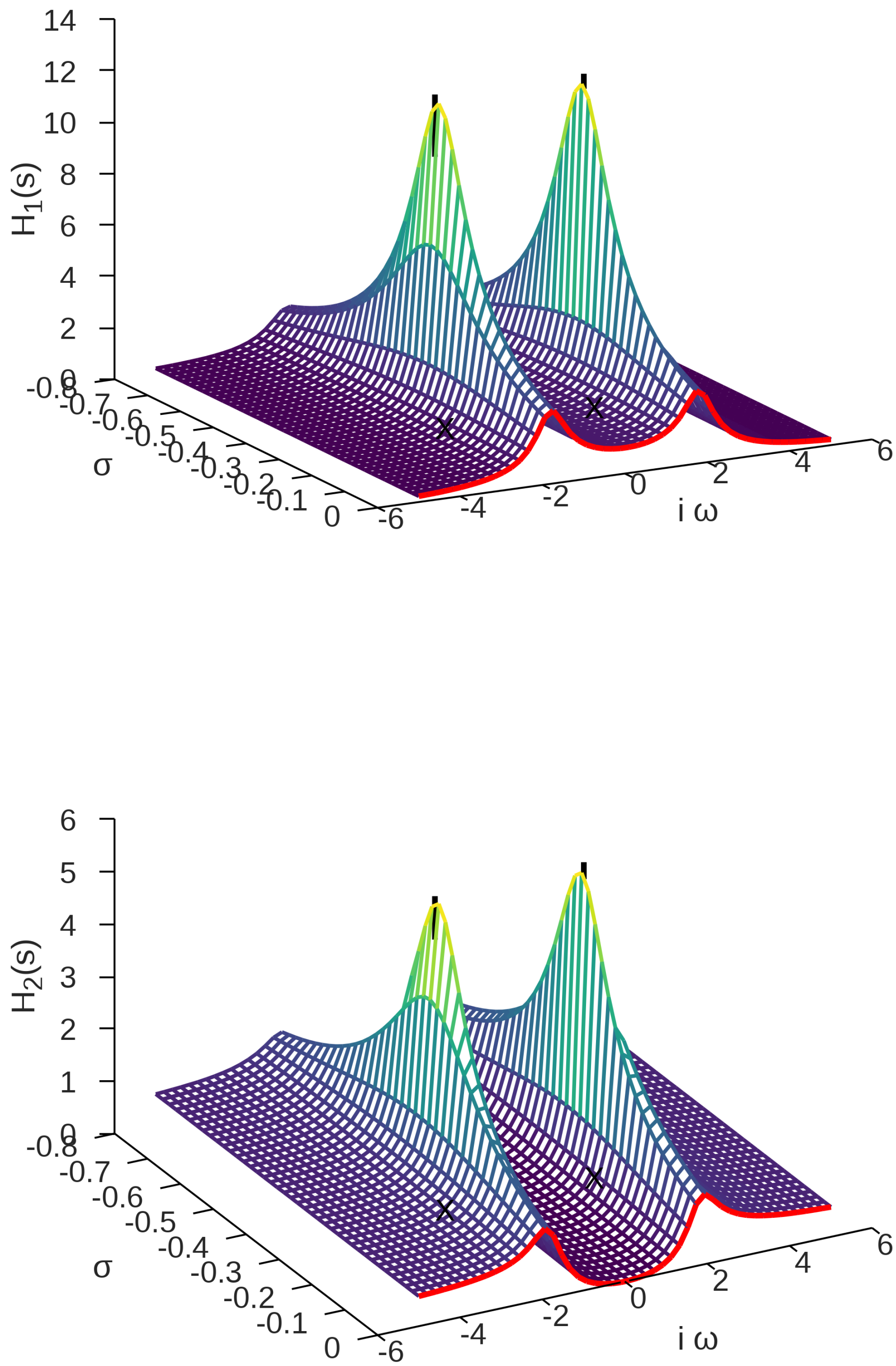

Figure 12. Laplace-domain transfer function of the linear time invariant system given in equations (156) and (157). Poles are marked with "x" on the Laplace plane ($s = \sigma + \mathrm{i}\omega$). The frequency response is shown as the red curve along the $\mathrm{i}\omega$ axis, at $\sigma = 0$. The real parts of the poles are all negative, meaning that the system is asymptotically stable.

$u(t)$, any linear time invariant system may be analyzed using the same sets of matlab commands.

40 Numerical Example 2: Butterworth Filters

The low-pass Butterworth filter of order n with cutoff frequency ω_c is defined by its transfer function magnitude

$$|H(s)|^2 = H(s)H(s^*) = \frac{b_0^2}{1 + (s/\omega_{\rm c})^{2n}}$$

The n poles of $H(s)$, at $s = p_k$, $(k = 1, 2, \ldots, n)$ are the n stable poles of the $2n$ roots of $1 + (p_k/\omega_c)^{2n} = 0$. Rearranging the characteristic equation,

$$\frac{p_k}{\omega_c} = (-1)^{1/(2n)} \ , \quad k \in \{1, 2, \ldots, n\} \quad \text{s.t.} \quad Re(p_k) < 0$$

The $2n$ roots of (-1) are found from $(\exp(\mathrm{i}\theta_k))^{2n} = -1$, which gives

$$\exp(2n \ \mathrm{i} \ \theta_k) = -1 \qquad \text{and} \qquad 2n\theta_k \in \{\pi, 3\pi, 5\pi, \ldots, (4n-1)\pi\} + q\pi \ .$$

where q is an integer. Solving for θ_k,

$$\theta_k = \pi \left(\frac{2k-1}{2n} + \frac{q}{2n} \right) \ , \qquad k \in \{1, 2, \ldots 2n\} \ .$$

The n stable poles correspond to $q = n$.

$$\begin{aligned} p_k &= \omega_c \ \left[\exp(2n \ \mathrm{i} \ \theta_k)\right]^{1/(2n)} \\ p_k &= \omega_c \ \exp\left[\mathrm{i}\pi \left(\frac{1}{2} + \frac{2k-1}{2n} \right) \right] \ , \qquad k \in \{1, 2, \ldots, n\} \end{aligned} \tag{158}$$

The characteristic equation $1 + (s/\omega_c)^{2n}$ (of order $2n$) may be approximated as an n-th order polynomial $a_0 + a_1 s + \ldots + a_{n-1}s^{n-1} + s^n$, that contains only the stable complex roots of $1 + (s/\omega_c)^{2n}$. At $s = 0$, $H(s) = b_0/a_0$. So setting $b_0 = a_0$ provides unity gain at $s = 0$. These coefficient values can be used in a companion matrix realization of a low-pass Butterworth filter.

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \tag{159}$$

$$y = \begin{bmatrix} a_0 & 0 & \cdots & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

in which the coefficients $a_0, ..., a_{n-1}$ are computed from the stable poles, $p_1, ..., p_n$.

The transfer function of the corresponding high-pass filter enforces $H(s) \to 0$ as $s \to 0$ by placing n zeros at $s = 0$.

$$H(s) = \frac{s^n}{a_0 + a_1 s + a_2 s^2 + ... + a_{n-1}s^{n-1} + s^n} \ ,$$

from which a companion matrix realization of a high-pass Butterworth filter is found to be

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u \tag{160}$$

$$y = \begin{bmatrix} -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + [1]u$$

in which the coefficients $a_0, ..., a_{n-1}$ are found from the same set of stable poles, $p_1, ..., p_n$, as used in the low pass filter. These continuous time realizations may be transformed to discrete time realizations with a first order hold. A state space implementation of Butterworth filters that incorporate the matrix exponential are significantly more stable than those based on the (approximate) bi-linear transformation.

41 References

1. Dennis S. Bernstein, *Matrix Theory, Facts and Formulas*, 2003
2. Bernstein, D.S. and Bhat, S.P., "Liapunov Stability, Semi Stability, and Asymptotic Stability of Matrix Second Order Systems," *ASME Journal of Mechanical Design,* 117 (1995): 145-153.
3. Erik Cheever, Linear Physical System Analysis, E12 course notes, Swarthmore College, https://lpsa.swarthmore.edu/, 2019.
4. C-T Chen, *Linear Systems Theory and Design,* 3rd ed. Oxford University Press, 1999.
5. C-T Chen, *Introduction to Linear System Theory,* Holt, Rienhart, and Winston, 1970
6. J. Doyle, B.A. Francis, and A. Tannenbaum, *Feedback Control Theory,* Prentice-Hall, 1990. pp 13–31.
   (recommended) http://www.duke.edu/~hpgavin/ce648/DFT.pdf
7. Gavin, H.P., Fourier Series, Fourier Transforms, and Periodic Response to Periodic Forcing, CEE 541 course notes, 2018
8. Gavin, H.P., Random Processes, Correlation, Power Spectral Density, Threshold Exceedance, CEE 541 course notes, 2018
9. T. Glad and L. Ljung, *Control Theory: Multivariable & Nonlinear Methods,* Taylor & Francis, 2000. pp 15–18, 63, 111.

10. H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems,* John Wiley & Sons, 1972. pp 97–111.

11. A.J.G. MacFarlane and N. Karcanias, "Poles and zeros of linear multivariable systems: A survey of the algebraic, geometric and complex-variable theory," *International Journal of Control*, 24(1):33-74, July 1976

12. J.M. Maciejowski, *Multivariable Feedback Design,* Addison-Wesley, 1989. pp 97–101.

13. C.B. Moler and G.W. Stewart, "An Algorithm for Generalized Matrix Eigenvalue Problems," *SIAM J. Numer. Anal.*, 10(2) (1973): 241-256.

14. H.H. Rosenbrock, "The zeros of a system," *Int'l J. Control*, 18(2) (1973): 297-299.

15. S. Skosgestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design,* John Wiley & Sons, 1996. pp 114–116, 514–526.

16. K. Zhou, J.C. Doyle, and K. Glover, *Robust and Optimal Control,* Prentice-Hall, 1995. pp 28–30, 91–104.