MA2011-W2 Dynamical Systems

```
clear all, disp(date)
```

```
18-Mar-2024
```

Zero-Order Systems

Here, by zero-order systems, we mean systems for which the output $y(t)$ at any given time t only depends on the current value of the input $x(t)$ but not on its history. These systems can be symply modeled by a, possibly nonlinear, *algebraic equation*, e.g.

$$y(t) = f(x(t), t)$$

The function $f(\cdot, t)$ might or might not depend explicitly on time t, in which case we denote the systems as *time-invariant*:

$$y(t) = f(x(t))$$

The simplest case is that of linear systems, i.e. those which can be simply described by a *linear equation*

$$y(t) = a(t) \cdot x(t) + b(t)$$

The easiest case is when the system is *Linear and Time-Invariant (LTI)*:

$$y(t) = a \cdot x(t) + b$$

i.e. when a and b are constant.

First-Order Systems (LTI) systems

Very often, the output of a system will depend also on the history of the input. Many systems of interest can be described via O*rdinary Differential Equations* (ODE). Besides its current input $u(t)$, the *state of the system* $x(t)$ will also depend on its time derivatives $\frac{d}{dt}x(t), \frac{d^2}{dt^2}x(t), \cdots, \frac{d^n}{dt^n}x(t)$, up to the *n*-th order.

First-oder systems only involve derivatives up the first order, i.e. it will only contain $\frac{d}{dt}x(t)$ (denoted as $\dot{x}(t)$, for short).

A *linear, time-invariant first-order system* can in general be described as

$$\dot{x}(t) + \frac{x(t)}{\tau} = u(t)$$

$$x(0) = x_0$$

where

- τ is the time constant;
- $u(t)$ is the forcing input;
- x_0 is the initial condition.

In MATLAB, this can be easily written as

```
syms x(t) u(t)
syms tau x0 real
assume (tau > 0)            %% NOTE this assumption
myEQ = diff(x) == -x/tau  + u      %% fist order ODE
```

```
myEQ(t) =
```

$$\frac{\partial}{\partial t}\, x(t) = u(t) - \frac{x(t)}{\tau}$$

```
myIC = x(0) == x0
```

myIC = $x(0) = x_0$

```
dsolve(myEQ, myIC)
```

```
ans =
```

$$e^{-\frac{t}{\tau}} \left(x_0 + \int_0^t e^{y/\tau}\, u(y)\, \mathrm{d}y \right)$$

A few words on Dynamical Analogies - the heart of mechatronics

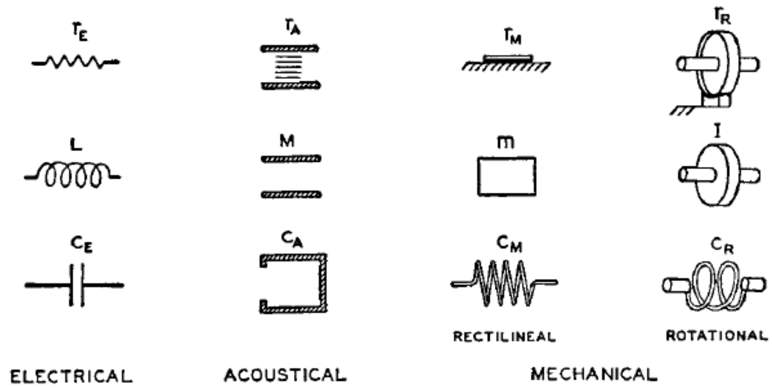


FIG. 2.1. Graphical representation of the three basic elements in electrical, mechanical rectilineal, mechanical rotational and acoustical systems.

r_E = electrical resistance	r_A = acoustical resistance	r_M = mechanical rectilineal resistance	r_R = mechanical rotational resistance
L = inductance	M = inertance	m = mass	I = moment of inertia
C_E = electrical capacitance	C_A = acoustical capacitance	C_M = compliance	C_R = rotational compliance

figure from [Olson 1958]

- Olson, H. F. (1958). *Dynamical analogies* (Vol. 2729). Princeton: Van Nostrand.

	electrical	mechanical (lin.)	mechanical (rot.)
flow	current [A]: i	lin. vel. [m/s]: $\dot{x}$	ang. vel [rad/s]: $\dot{\theta}$
effort	voltage [V]: Δv	force [N]: f	torque [Nm]: τ
power [W]	$\Delta v \cdot i$	$f \cdot \dot{x}$	$\tau \cdot \dot{\theta}$
energy [J]	$\int \Delta v \cdot i \, dt$	$\int f \cdot \dot{x} \, dt$	$\int \tau \cdot \dot{\theta} \, dt$
Elements			
resistance	$\Delta v = R\, i$	$f = b\, \dot{x}$	$\tau = \beta\, \dot{\theta}$
inertia	$\Delta v = L\, \frac{di}{dt}$	$f = m\, \ddot{x}$	$\tau = I\, \ddot{\theta}$
compliance	$\Delta v = \frac{1}{C} \int i \, dt \equiv \frac{q}{C}$	$f = \int k\dot{x} \, dt \equiv k\, x$	$\tau = \int \kappa\dot{\theta} \, dt \equiv \kappa\, \theta$
En. storage			
kinetic energy	$E = \frac{1}{2}Li^2$	$E = \frac{1}{2}m\dot{x}^2$	$E = \frac{1}{2}I\dot{\theta}^2$
potential energy	$E = \frac{1}{2}C\Delta v^2$	$E = \frac{1}{2}k\Delta x^2$	$E = \frac{1}{2}\kappa\Delta\theta^2$

DC Analysis (response to constant inputs)

Very often, a system is subjected to a constant input

$$\dot{x}(t) + \frac{x(t)}{\tau} = u_0$$

$$x(0) = x_0$$

```
syms u0 real

myEQ0 = subs(myEQ, u(t), u0)
```

myEQ0(t) =

$$\frac{\partial}{\partial t} x(t) = u_0 - \frac{x(t)}{\tau}$$

```
DC_Sol = dsolve(myEQ0, myIC)
```

```
DC_Sol =
```

$$\tau u_0 + e^{-\frac{t}{\tau}} (x_0 - \tau u_0)$$

Let's see what happens when $t \to \infty$, let's define

$$x_\infty := \lim_{t \to \infty} x(t)$$

```
limit(DC_Sol, t, inf)
```

ans = τu_0

From this, we shall rewrite the first order equation as

$$\dot{x}(t) = \frac{x_\infty - x(t)}{\tau}$$

$$x(0) = x_0$$

```
syms x_inf real
myEQ0 = subs(myEQ, u(t), x_inf/tau)
```

```
myEQ0(t) =
```

$$\frac{\partial}{\partial t} x(t) = \frac{x_{\mathrm{inf}}}{\tau} - \frac{x(t)}{\tau}$$

```
dsolve(myEQ0, myIC)
```

```
ans =
```

$$x_{\mathrm{inf}} + e^{-\frac{t}{\tau}} (x_0 - x_{\mathrm{inf}})$$

and we wish to know how it will respond. In this case, rather than a generic forcing input, we should consider a constant input $u(t)$ and a system initially in its *zero* state, i.e. $x(0) = 0$

```
mylegend = {};
for tau_list = [0.1 0.2 1 2 5 10]
    SOL = dsolve (subs(myEQ0, {x_inf, tau}, {5, tau_list}), 'x(0)==0')
    fplot(SOL, [0 10]); grid on; hold on
    mylegend= {mylegend{:}, ['\tau = ' num2str(tau_list) ' sec']};
    xlabel('time [sec]'); ylabel('x(t) [V]')
    title('x_\infty = 5V')
end
```

SOL = $5 - 5\, e^{-10\, t}$

SOL = $5 - 5\, e^{-5\, t}$

SOL = $5 - 5\,e^{-t}$

SOL =

$$5 - 5\,e^{-\frac{t}{2}}$$

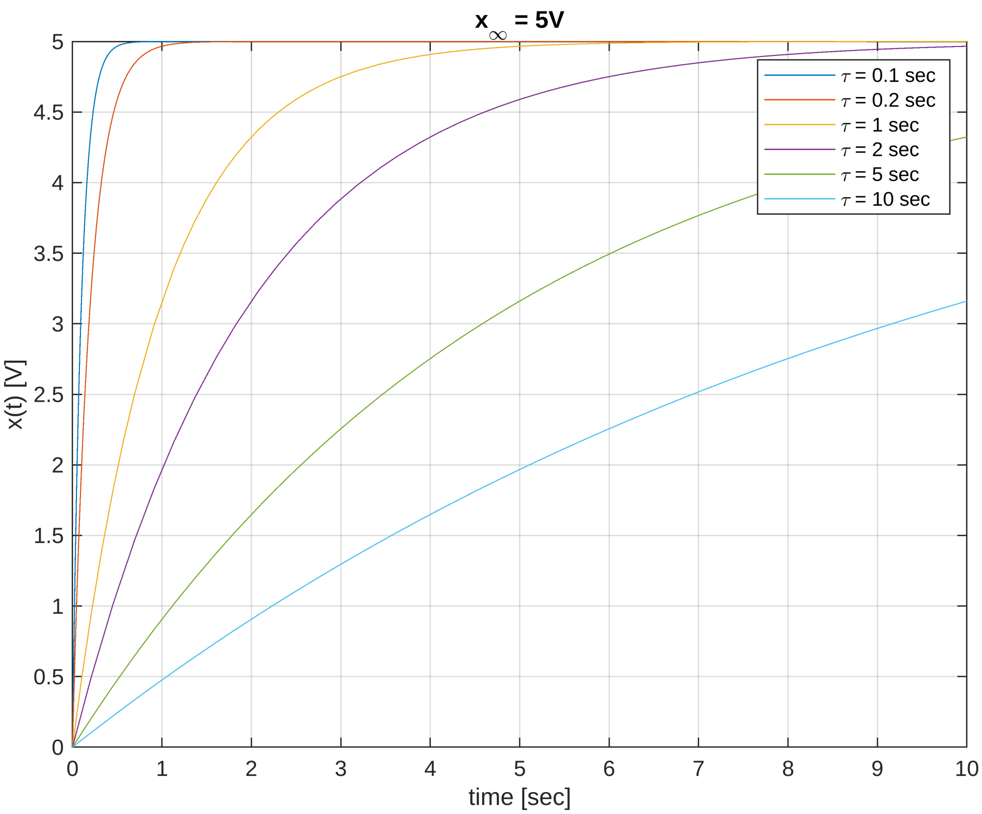
SOL =

$$5 - 5\,e^{-\frac{t}{5}}$$

SOL =

$$5 - 5\,e^{-\frac{t}{10}}$$

```
legend(mylegend);
```



To manually sketch exponential responses given x_0, x_∞, τ it is useful to evalutate the time derivative of the explonential at time $t = 0$, i.e.

$$\dot{x}(0) = \frac{x_\infty - x_0}{\tau}$$

```
subs(myEQ0, t, 0)
```

ans(t) =

$$\left(\left(\frac{\partial}{\partial t}\ x(t)\right)\bigg|_{t=0}\right) = \frac{x_{\mathrm{inf}}}{\tau} - \frac{x(0)}{\tau}$$

as this provides the slope of the tangent line

```
tau_val = 2;
x_0_val = 3;
x_inf_val = 7;
range = [0 3*tau_val];
MyExp = subs( dsolve(myEQ0, 'x(0)==x0'), {x0, x_inf, tau}, {3, 7, tau_val})
```

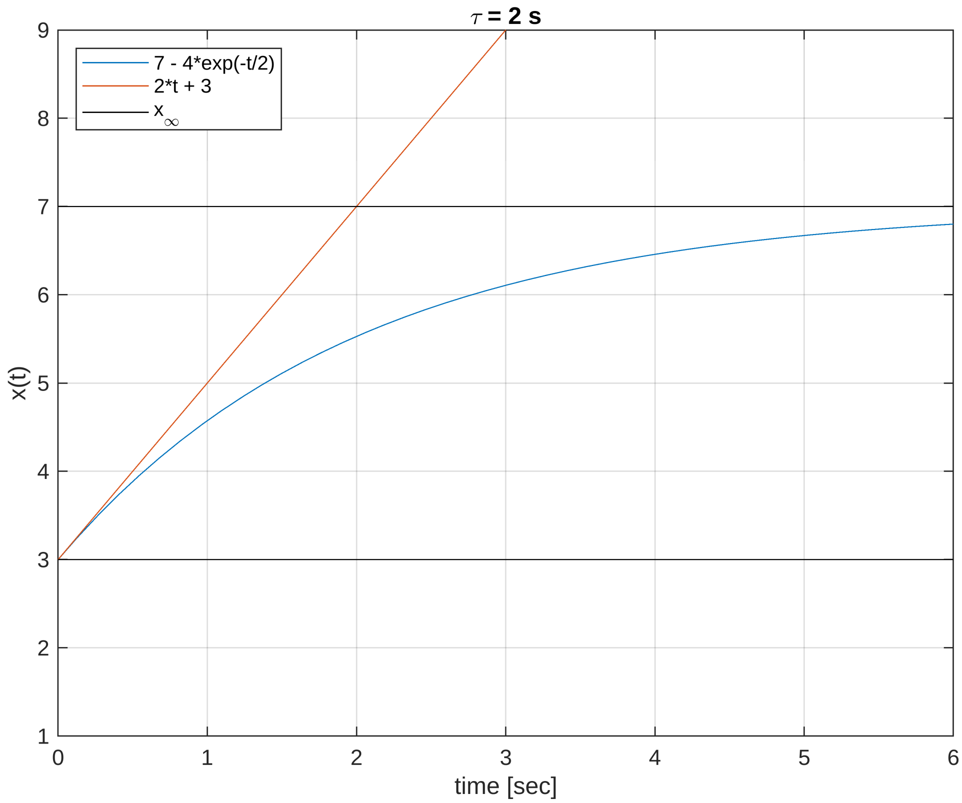
MyExp =

$$7 - 4\,\mathrm{e}^{-\frac{t}{2}}$$

```
MyTaylor = taylor (MyExp, t, 'Order', 2)
```

MyTaylor = $2\,t + 3$

```
figure
fplot(MyExp, range); grid on, hold on
fplot(MyTaylor, range)
fplot(x_inf_val, 'k', range)
fplot(x_0_val, 'k', range)
ylim([x_0_val-2 x_inf_val+2])
title(['\tau = ' num2str(tau_val) ' s'])
legend(char(MyExp), char(MyTaylor), 'x_\infty', 'Location', 'NorthWest')
 xlabel('time [sec]'); ylabel('x(t)')
```



AC Steady-State analysis (steady-state response to sinusoidal inputs)

AC-analysis is based on the fact that given a purely sinusoidal input, e.g. $\cos(\omega t)$, after a long time, we expect to see a purely sinusoidal output, e.g. $A_\omega \cos(\omega t + \phi_\omega)$

- at the same frequency ω
- but with likely different (frequency-dependent) Amplitude A_ω and phase ϕ_ω

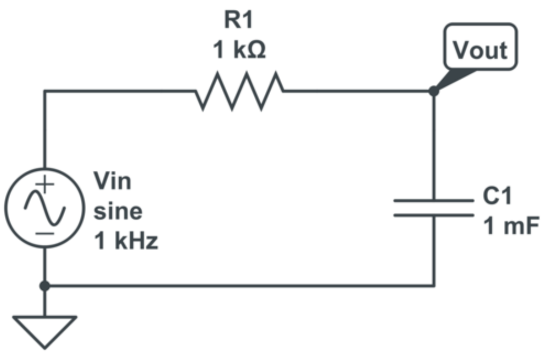
$$\cos(\omega t) \Rightarrow \|system\| \Rightarrow A_\omega \cos(\omega t + \phi_\omega)$$

The **objective of AC analysis** is to determine Amplitude A_ω and phase ϕ_ω for every input frequency ω

This can be accomplished in basically two ways

- via trigonometric manipulation (hard way)
- via complex exponentials (easy way)

Let's look at a simple case, a 1st order RC circuit



The hard way - via trigonometry

```
C = 1e-3;        %% F
R = 1e3;         %% Ohms
syms phi t omega real
syms A positive     %% if we don't impose a positve value, we will get two
solutions later on
V_in = cos(omega * t)
```

V_in = $\cos(\omega\ t)$

```
V_out = A * cos(omega *t + phi)
```

V_out = $A\ \cos(\phi + \omega\ t)$

Define device equations

```
i = C* diff(V_out)  %% the command 'diff' takes the time derivative
```

i =

$$-\frac{A\ \omega\ \sin(\phi + \omega\ t)}{1000}$$

```
V_R = R* i
```

V_R = $-A\ \omega\ \sin(\phi + \omega\ t)$

Solve equations $\forall t$, in fact, only for a couple of values, then we can verify that solutions are valid at all times.

```
KVL = V_in == V_R + V_out
```

KVL = $\cos(\omega\ t) = A\ \cos(\phi + \omega\ t) - A\ \omega\ \sin(\phi + \omega\ t)$

```
expand (KVL)
```

ans = $\cos(\omega\ t) = A\ \cos(\omega\ t)\cos(\phi) - A\ \sin(\omega\ t)\sin(\phi) - A\ \omega\ \cos(\omega\ t)\sin(\phi) - A\ \omega\ \sin(\omega\ t)$

```
myEquations =  [subs(KVL, t, 0); expand(subs(KVL, t, 1/omega*pi/2))]
```

```
myEquations =
```

$$\begin{pmatrix} 1 = A\cos(\phi) - A\,\omega\sin(\phi) \\ 0 = -A\sin(\phi) - A\,\omega\cos(\phi) \end{pmatrix}$$

```
myVariables = [A; phi]
```

```
myVariables =
```

$$\begin{pmatrix} A \\ \phi \end{pmatrix}$$

Note that to solve this trigonometric system, you will need to make use of identities such as

```
cos(atan(omega))
```

```
ans =
```

$$\frac{1}{\sqrt{\omega^2 + 1}}$$

```
sin(atan(omega))
```

```
ans =
```

$$\frac{\omega}{\sqrt{\omega^2 + 1}}$$

Or ask a symbolic to solve it for you...

```
SOL = solve(myEquations, myVariables)
```

```
SOL = struct with fields:
      A: 1/(omega^2 + 1)^(1/2)
    phi: -2*atan(((omega^2 + 1)^(1/2) - 1)/omega)
```

```
SOL.A, SOL.phi
```

```
ans =
```

$$\frac{1}{\sqrt{\omega^2 + 1}}$$

```
ans =
```

$$-2\,\operatorname{atan}\left(\frac{\sqrt{\omega^2 + 1} - 1}{\omega}\right)$$

NOTE: that $-2\tan^{-1}\left(\frac{\sqrt{\omega^2+1}-1}{\omega}\right) \equiv -\tan^{-1}\omega$

<u>However, this is a very cumbersome (too much trigonometry) way to proceed!!!</u>

The easy way via the 'complex trick'

again, start from the the time-domain equations, for simplicity let's set $\tau = 1$

```
myEQ = subs(myEQ, tau, 1)
```

myEQ(t) =

$$\frac{\partial}{\partial t}\, x(t) = u(t) - x(t)$$

Now, let's consider **complex exponentials** as inputs

$$e^{j\omega t} \Rightarrow \left|\overset{LTI}{SYS}\right| \Rightarrow A_\omega\, e^{j(\omega t + \phi_\omega)} = \underbrace{A_\omega\, e^{j\phi}}_{H(j\omega)} e^{j\omega t}$$

```
syms H_0 complex
syms omega real
j=sqrt(-1);
myEQ_AC = subs(myEQ, {u(t), x(t)}, {exp(j*omega *t), H_0 * exp(j*omega *t)})
```

myEQ_AC(t) = $H_0\, \omega\, \mathrm{e}^{\omega\, t\, \mathrm{i}}\, \mathrm{i} = \mathrm{e}^{\omega\, t\, \mathrm{i}} - H_0\, \mathrm{e}^{\omega\, t\, \mathrm{i}}$

```
eqn_cmplx = simplify(myEQ_AC)
```

eqn_cmplx(t) = $H_0\, (1 + \omega\, \mathrm{i}) = 1$

```
% simplify (real(AC_Sol_complex) - AC_Sol)
Sol_myEQ_AC = solve(eqn_cmplx, H_0)
```

Sol_myEQ_AC =

$$\frac{1}{1 + \omega\, \mathrm{i}}$$

Bode Plots

```
freq = logspace(-2, 2, 100);
H = subs(Sol_myEQ_AC,{ tau, omega}, {1, 2*pi*freq} );
figure
subplot(2,1,1)
loglog(freq, abs(H)); grid on; ylabel('amplitude'); title ('Bode Plots')
hold on
subplot(2,1,2)
```

```
semilogx(freq, angle(H)); grid on; xlabel('frequency [Hz]'); ylabel('phase
[rad]')
hold on
```

Cut-off frequency: the frequency at which the output is reduced by -3dB (or $1/\sqrt{2}$), clearly for a first order system

```
omega_0 = subs(1/tau, tau, 1)
```

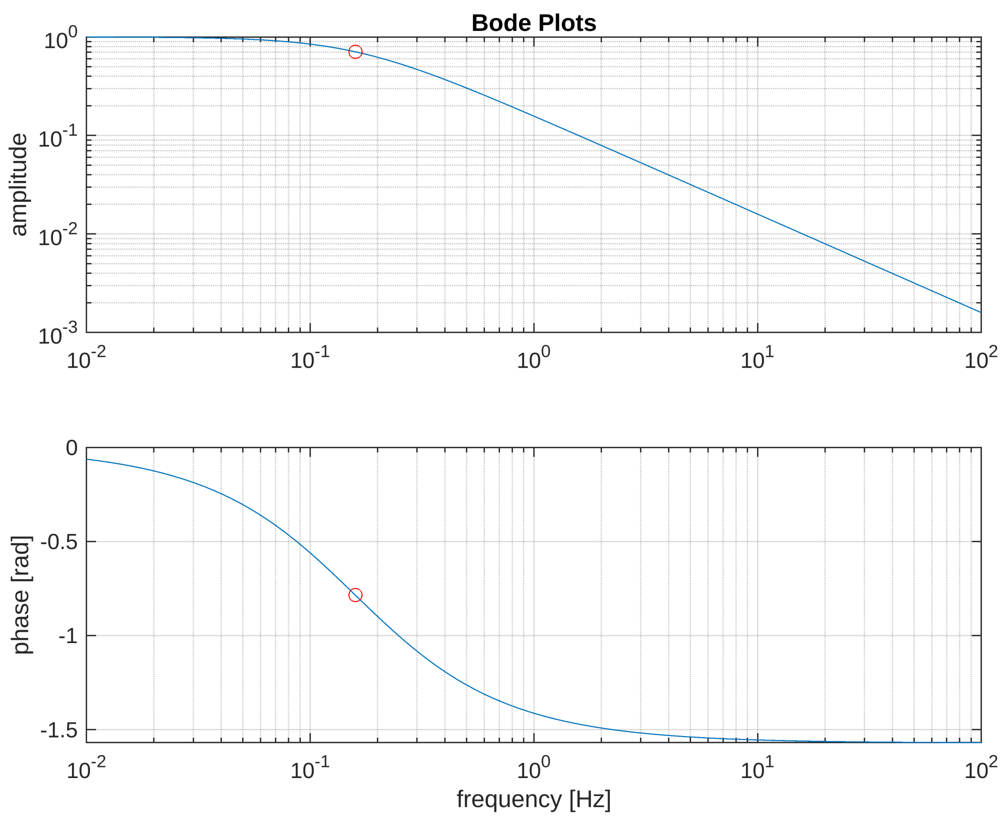
omega_0 = 1

```
f_0 = omega_0/2/pi
```

f_0 =

$$\frac{1}{2\,\pi}$$

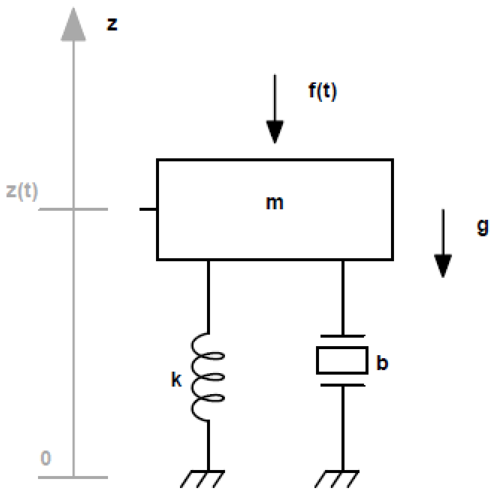
```
H_0 = subs(Sol_myEQ_AC,{tau, omega}, {1, 1} );
subplot(2,1,1)
loglog(omega_0/2/pi,  abs(H_0) , 'ro')
subplot(2,1,2)
semilogx(omega_0/2/pi,  angle(H_0) , 'ro')
```



2nd order dynamics

Newton's law

Consider a mass-springer-damper case:



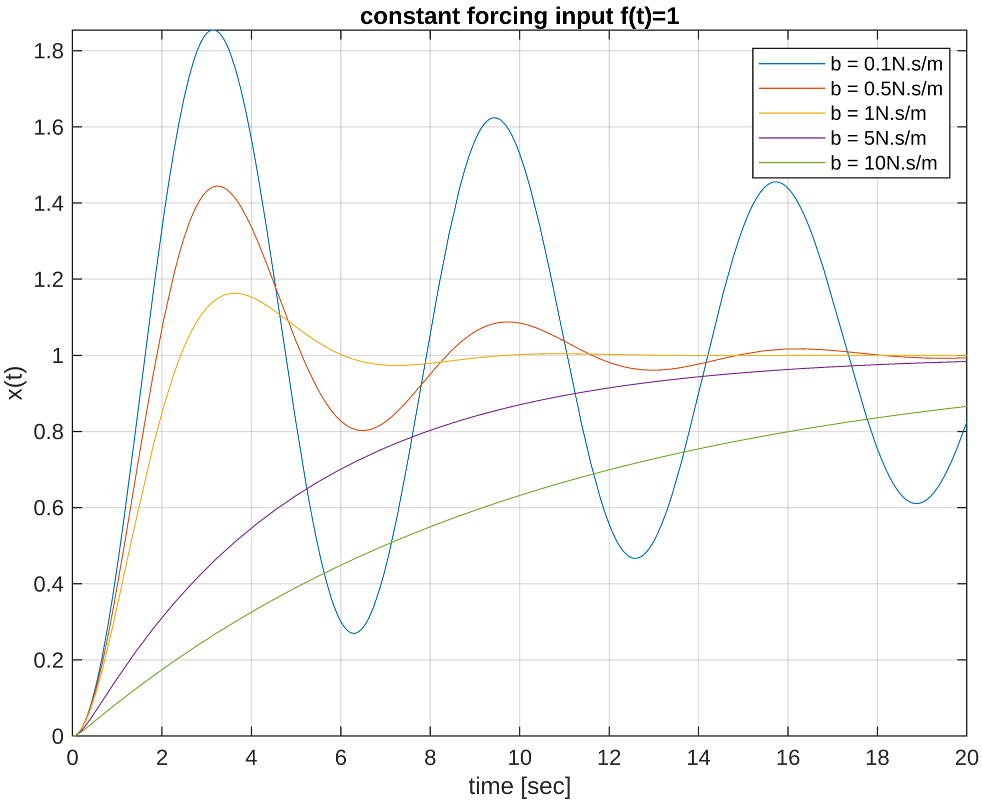

```
syms m b k real
syms f(t) x(t)
eqn_Newton = f(t) == m*diff(x(t), 2)+b*diff(x(t)) + k*x(t)
```

eqn_Newton =

$$f(t) = m\,\frac{\partial^2}{\partial t^2}\,x(t) + b\,\frac{\partial}{\partial t}\,x(t) + k\,x(t)$$

Time Response to constant forcing Input

```
figure;
mylegend = {};
for b_var = [0.1, 0.5, 1, 5, 10]
    mylegend = {mylegend{:}, ['b = ' num2str(b_var) 'N.s/m']};
    eqn_Newton_DC = subs(eqn_Newton, {f(t), m, b, k}, {1, 1, b_var, 1});
    Dx = diff(x);
    cond = [x(0)==0, Dx(0) == 0];
    Sol_Newton_time = dsolve(eqn_Newton_DC, cond);
    fplot(Sol_Newton_time, [0 20]); grid on; hold on
end
xlabel('time [sec]'); ylabel('x(t)'); title ('constant forcing input f(t)=1')
legend(mylegend)
```



AC Analysis

```
syms omega real
syms F_0 X_0 complex
AC_Exp_Newton = subs(eqn_Newton, {f(t), x(t)}, {F_0*exp(j*omega*t),
X_0*exp(j*omega*t)})
```

AC_Exp_Newton = $F_0\, \mathrm{e}^{\omega\, t\, \mathrm{i}} = X_0\, k\, \mathrm{e}^{\omega\, t\, \mathrm{i}} + X_0\, b\, \omega\, \mathrm{e}^{\omega\, t\, \mathrm{i}}\, \mathrm{i} - X_0\, m\, \omega^2\, \mathrm{e}^{\omega\, t\, \mathrm{i}}$

```
simplify( AC_Exp_Newton )
```

ans = $X_0\, m\, \omega^2 + F_0 = X_0\, (k + b\, \omega\, \mathrm{i})$

```
Sol_ODE2= solve(simplify (AC_Exp_Newton) , X_0)
```

Sol_ODE2 =

$$\frac{F_0}{-m\, \omega^2 + \mathrm{i}\, b\, \omega + k}$$

```
freq = logspace(-2, 1, 100);
H = subs(Sol_ODE2,{F_0, m, b, k, omega}, {1, 1, .1, 1, 2*pi*freq} );
figure
subplot(2,1,1)
```

```
loglog(freq, abs(H)); grid on; ylabel('amplitude'); title ('Bode Plots')
subplot(2,1,2)
semilogx(freq, angle(H)); grid on; xlabel('frequency [Hz]'); ylabel('phase
[rad]')
```