MA2011-W2 Dynamical Systems

```
date
```

```
ans = 
'01-Nov-2020'
```

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 =

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

Response to a constant input

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 =

$$\tau\, u_0 + \mathrm{e}^{-\frac{t}{\tau}} \left(x_0 - \tau\, u_0 \right)$$

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 = $\tau\, u_0$

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 =

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

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)')
    title('x_\infty = 5')
end
```

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

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

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

SOL =

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

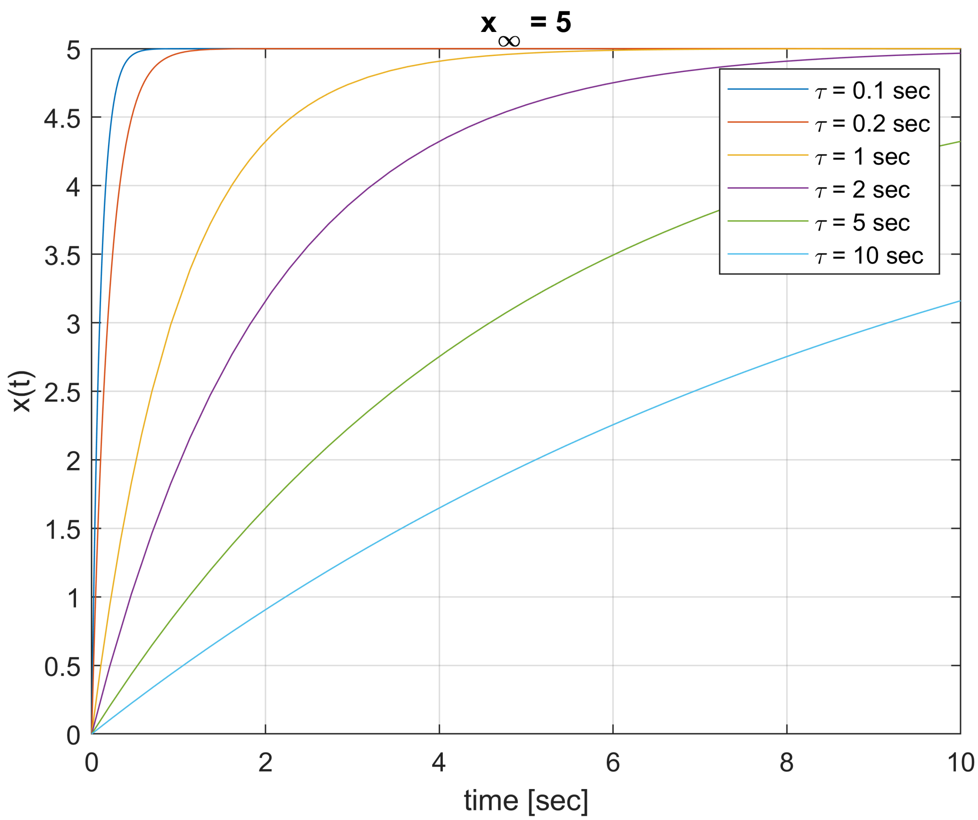
SOL =

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

SOL =

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

```
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) =

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

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})
```

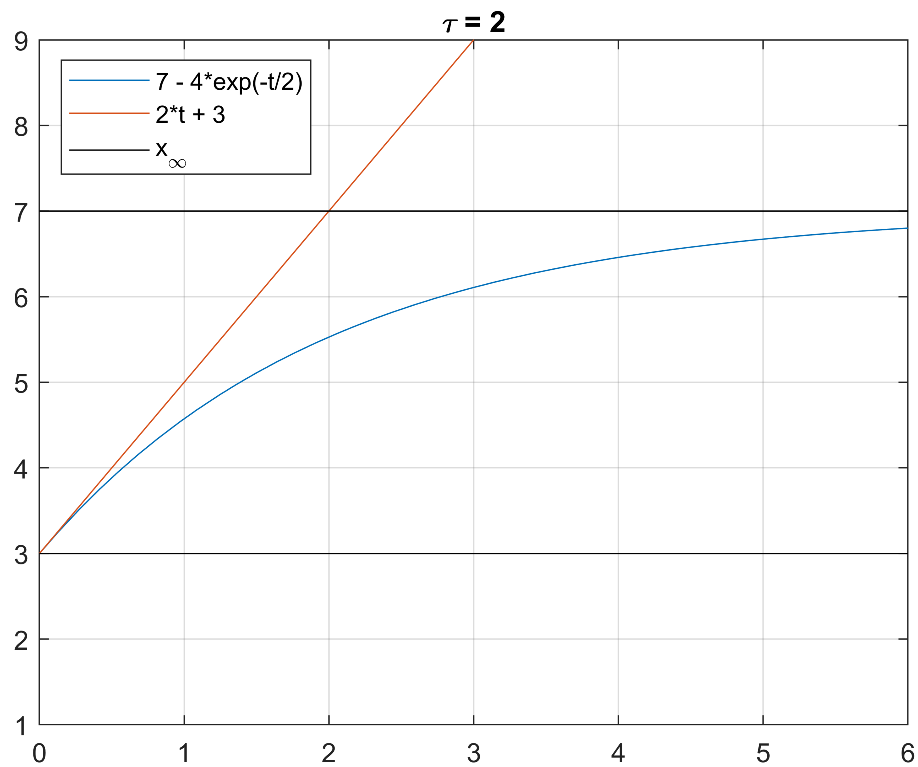
MyExp =

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

```
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)])
legend(char(MyExp), char(MyTaylor), 'x_\infty', 'Location', 'NorthWest')
```



AC analysis of 1st order systems via the 'complex trick'

```
syms U_0 X_0 complex
syms omega real
j=sqrt(-1);
myEQ_AC = subs(myEQ, {u(t), x(t)}, {U_0 * exp(j*omega *t), X_0 * exp(j*omega *t)})
```

myEQ_AC(t) =

$$X_0\,\omega\,e^{\omega\,t\,i}\,i = U_0\,e^{\omega\,t\,i} - \frac{X_0\,e^{\omega\,t\,i}}{\tau}$$

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

eqn_cmplx(t) = $X_0\,(1+\omega\,\tau\,i) = U_0\,\tau$

```
% simplify (real(AC_Sol_complex) - AC_Sol)
```

$V_R = R \cdot I = RC \cdot A\omega\, e^{j(\omega t + \varphi)}$
$= CR \cdot A\omega \cdot e^{j\varphi} \cdot e^{j\omega t}$

```
Sol_myEQ_AC = solve(eqn_cmplx, X_0)
```

Sol_myEQ_AC =

$$\frac{U_0\,\tau}{1+\omega\,\tau\,\mathrm{i}}$$

KVL : $V_{in} = V_R + V_{out}$

$\cancel{e^{j\omega t}} = RC \cdot A\omega \cdot e^{j\varphi} \cdot \cancel{e^{j\omega t}} + A \cdot e^{j\varphi} \cdot \cancel{e^{j\omega t}}$

Bode Plots

```
freq = logspace(-2, 2, 100);
H = subs(Sol_myEQ_AC,{U_0, tau, omega}, {1 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 =

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

$H(j\omega) = \dfrac{1}{1+j\omega RC}$

```
H_0 = subs(Sol_myEQ_AC,{U_0, tau, omega}, {1 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')
```

Plot $|H(j\omega)|$ and $\measuredangle H(j\omega)$

$\dfrac{1}{|a+jb|} = \dfrac{1}{\sqrt{a^2+b^2}}$ $\quad \measuredangle\, a+jb = \tan^{-1}\left(\dfrac{b}{a}\right)$

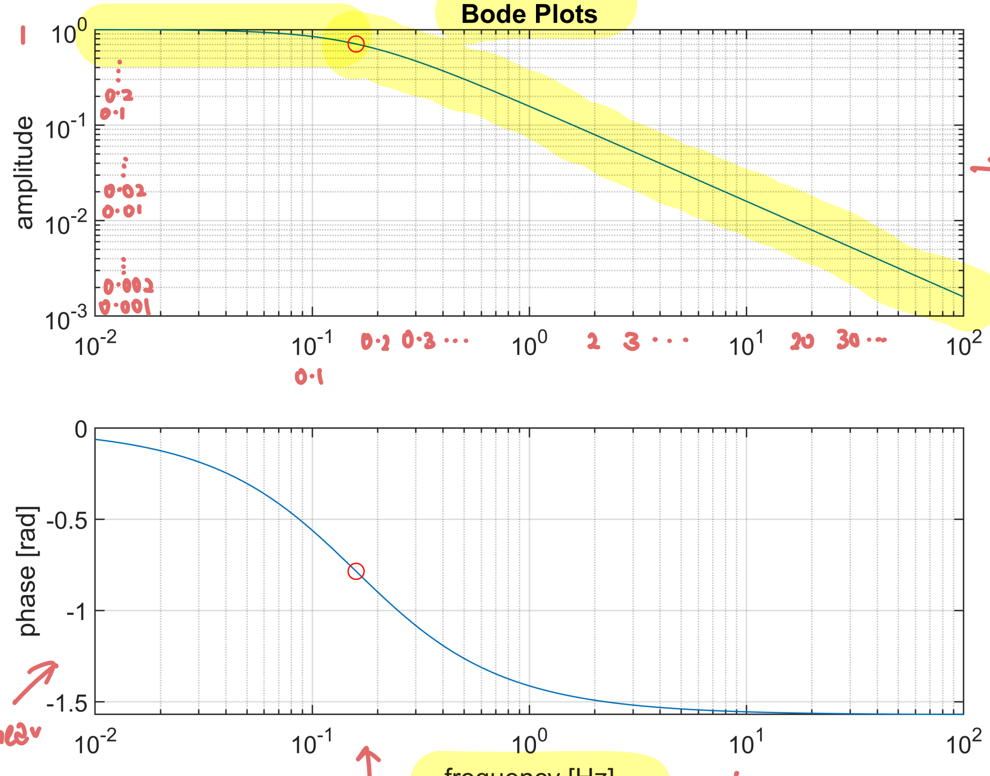
$\measuredangle \dfrac{1}{a+jb} = -\tan^{-1}\left(\dfrac{b}{a}\right)$

$|H(j\omega)| = \dfrac{1}{\sqrt{1+\omega^2R^2C^2}}$

$\measuredangle H(j\omega) = -\tan^{-1}(\omega RC)$

logarithmic axis

RC = 1



→ To appreciate low & high frequency

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

* know how to read

$-\tan^{-1}(w)$

linear

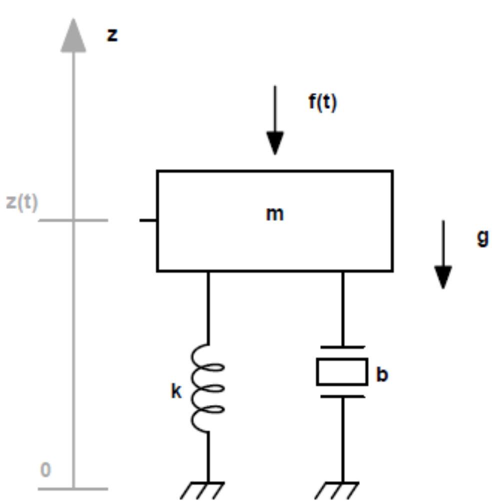
logarithmic

semi-logarithmic

2nd order dynamics

Newton's law

Consider a mass-springer-damper case:



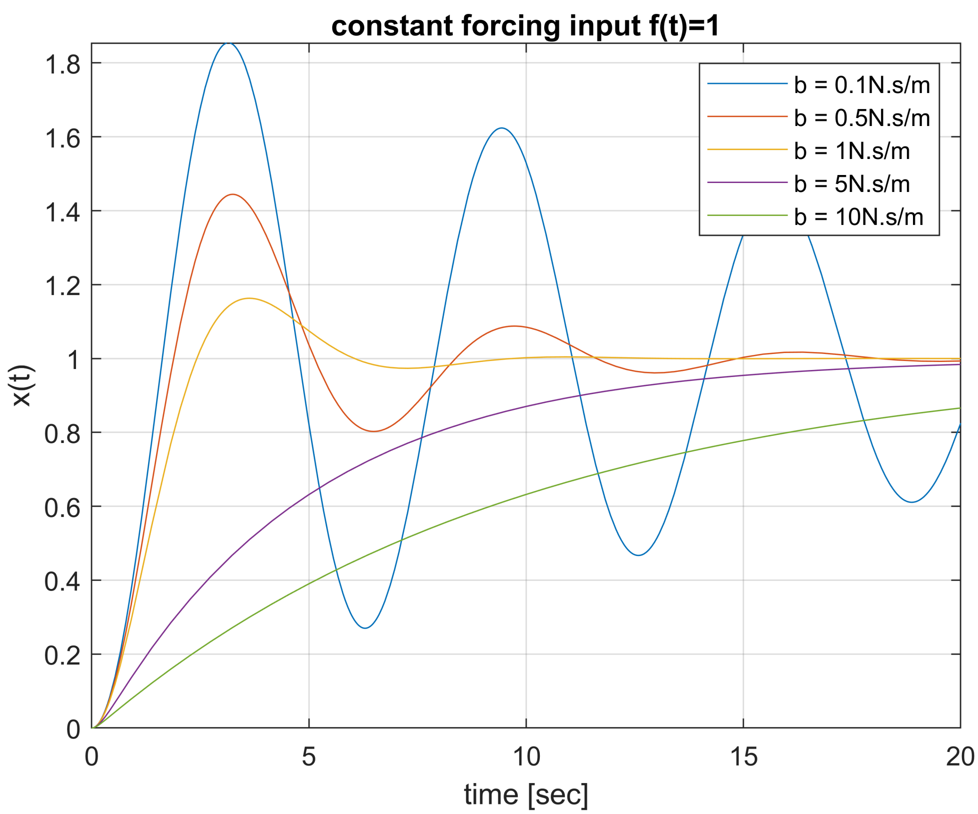

```
syms m b k real
syms f(t) x(t)
eqn_Newton = f(t) == m*diff(x(t), 2)+b*diff(x(t)) + k*x(t)
```

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

```
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]')
```