Matlab codes:

1. Finding laplace

syms t , s

laplace (something in way of computer)

Finding laplace inverse.

syms (s, t)

ilaplace (" ")

2. Finding transfer function: $G(s) = \frac{Y(s)}{V(s)}$

num = [ ] coefficients of numerator taking into consideration anything equalling 0.

den = [ ] coefficients of denominator

sys = tf (num, den).

3. Finding zero-pole gain (ZPK): $G(s) = \frac{Y(s)}{V(s)}$

sys1 = zpk(, [ ],)

s = what in numerator

s = what in denominator

any coefficients outside

or

s = tf {'s'}

G = (something in way of computer) ← newer matlab.

Scanned with CamScanner

4. Finding State-Space models: $G(s) = \frac{Y(s)}{U(s)}$

① $\frac{Y(s)}{U(s)} = \frac{Y}{R} = \frac{X}{R} \times \frac{Y}{X}$

② $R = \text{coef}\, \ddot{X} + \text{coef}\, \dot{x} + \text{coef}\, X$

$Y = \text{coef}\, \dot{x} + \text{coef}\, x$.

$x_1 = x, \quad \boxed{x_2 = \dot{x}}$ *

$R = \text{coef}\, \dot{x}_2 + \text{coef}\, x_2 + \text{coef}\, x_1$

$\boxed{\dot{x}_2 = -\text{coef}\, x_2 + -\text{coef}\, x_1 + R}$ *

③ $R = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\text{coef} & -\text{coef} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} R$

$Y = [\text{coef} \quad \text{coef}] \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]\, R$.

④ sys = ss([0 1 ; -coef -coef],
[0 ; 1], [coef coef],
0)

5. finding time response of systems: G(s) (plotting)

impulse (G(s))

step (G(s))

lsim (G(s)) (general time response)

conv (G(s)) (polynomial multiplication)

deconv (G(s)) (polynomial division).

residue(G(s)) (partial fraction expansion)

6. finding root locus · G(s)

rlocus(G (s))

<u>or</u>

numerator coef. denominator, [illegible] coeff.

rlocus(tf([ ], conv(conv[ ],[ ]

[ ])))

7. sisotool.

sisotool (G(s))

remember that:

$As+B$ [A B]

As^2+Bs+C [A B C]

As^3+Bs^2+Cs+D [A B C D]

if anything of those is missing, its 0

1.OD ... tial equations:

Using numeric way to solve differential equ. in Matlab:

[1] A simple 1st order that has given the parameters:

function dx = mydiff (t, x)
(t → time)

first part

b = 1 ;
p = 0.5 ;] given parameters

dx = b * x - p * x ^ 2 ; diff. equ.

tspan = [0, 1] time span (from 0 - 1 hour)

x_0 = 100 ; starting value.

[t , y] = ode 45 (@mydiff, tspan , x_0) ;

plot (t , y)

* tip : always have $\dot{x}$ موضوع للقانون before you begin.

* note : we also have ode23

[2] 1st order that has unknown parameters.

function dx = mydiff (t , x, param)
added to define a and b

a = param (1) ; unknown
b = param(2) ; unknown

dx = a * x + b ;

tspan = [0 5] ;

x_0 = 1 ;

a = $-\frac{1}{5}$; given as $-\frac{1}{T}$

b = 1 ; to make it easy to change.

param [a b] ; التكرار يفيد التأكيد

[t , y] = ode 45(@mydiff , tspan, x_0, [], param).
no idea why this is here.

plot (t,y)

3 Not so simple second order (given parameters)

1. begin by making $\ddot{x}$ موضوع القانون

2. we cant have $\ddot{x}$, so $x_2 = \dot{x}$, $x_1 = x$ (1)

after putting those in original it is (2)

3. function dx = mydiff (t, x)

```
[m,n] = size(x);
dx = zeros(m,n)
```

I have no idea why we put these.

```
dx(1) = x(2);   (1)
dx(2) = (2-2*t*x(2)-3*x(1))/(1+t^2);   (2)

tspan [0  5];
x0 = [0 ; 1];

[t,x] = ode23(@mydiff, tspan, x0);
plot(t,x)
legend('x1', 'x2')
```

```
tspan [0, 5];
x0 = [0 ; 1];
[t,x] = ode23(@mydiff, tspan, x0);
```

plot(t, x(:,2))

Symbolic way to solve differential equations:

① $\frac{dy}{dt} = ay.$

syms y(t) a → a variable

equation

eqn = diff(y, t) == a*y; → $\frac{dy}{dt}$

S = dsolve(eqn)

solution diff.

② $\frac{dy}{dt} = z \qquad \frac{dz}{dt} = -y$ (system)

syms y(t) z(t)

eqns = [diff(y,t) == z, diff(z,t) == -y];

S = dsolve(eqns) ⇒ without assignment

or

[ysol(t), zsol(t)] = dsolve(eqns) ⇒ with assignment

③ $\frac{dy}{dx} = \frac{1}{x^2} e^{-\frac{1}{x}}$

```
syms y(x)
eqn = diff(y) == exp(-1/x) / x^2;
ysol(x) = dsolve(eqn)
```
⇒ without specifying initial condition

or

```
cond = y(0)==1;
S = dsolve(eqn, cond)
```
⇒ specifying initial condition to eliminate constants from solutions.

④ $\frac{d^2y}{dx^2} = a^2 y$, $y(0) = b$, $y'(0) = 1$

```
syms y(t) a b
eqn = diff(y, t, 2) == a^2 * y;
Dy = diff(y,t);
cond = [ y(0) == b, Dy(0) == 1];
ysol(t) = dsolve(eqn, cond).
```
second order

assigning diff(y,t) to Dy

conditions