Importing the necessary libraries

In [101…
```
import sympy as sym
```

To find the **Extremal Trajectory** for functional

$$J = \int_0^{t_f} \{2(\dot{x}(t))^2 + 24tx(t)\}dt$$

given the **initial condition** $x(0) = 0$ and the **final condition** $x(t_f) = 2$.

Let $x(t) = x$ and $\dot{x}(t) = \dot{x}$, so

$$V(.) = 2\dot{x}^2 + 24tx$$

Partially Differentiate with respect to x and $\dot{x}$

$$\frac{\delta V}{\delta x} = 24t$$

$$\frac{\delta V}{\delta \dot{x}} = 4\dot{x}$$

Solving we get 2 optimized trajectories

$$x^*(t) = t^3 - 1.117t$$

$$x^*(t) = t^3 + 2.687t$$

We need to compute 2 quantities

Hessian Matrix H and Second Variation $\delta^2 J$

Hessian Matrix

$$H = \begin{bmatrix} \frac{\delta^2 V}{\delta x^2} & \frac{\delta^2 V}{\delta x \delta \dot{x}} \\ \frac{\delta^2 V}{\delta \dot{x} \delta x} & \frac{\delta^2 V}{\delta \dot{x}^2} \end{bmatrix}$$

Second Variation

$$\delta^2 J = \frac{1}{2}\int_{t_0}^{t_f}\{\begin{bmatrix}\delta x & \delta\dot{x}\end{bmatrix}\begin{bmatrix}\frac{\delta^2 V}{\delta x^2} & \frac{\delta^2 V}{\delta x \delta\dot{x}} \\ \frac{\delta^2 V}{\delta\dot{x}\delta x} & \frac{\delta^2 V}{\delta\dot{x}^2}\end{bmatrix}\begin{bmatrix}\delta x \\ \delta\dot{x}\end{bmatrix}\}dt$$

$$\implies \delta^2 J = \frac{1}{2}\int_{t_0}^{t_f}\{\begin{bmatrix}\delta x & \delta\dot{x}\end{bmatrix} H \begin{bmatrix}\delta x \\ \delta\dot{x}\end{bmatrix}\}dt$$

Compute Hessian Matrix

In [102…
```
# Initialise the symbolic variables

x = sym.symbols('x')
x_dot = sym.symbols('x_dot')
t = sym.symbols('t')

V = sym.symbols('V')

# Define the symbolic equations

V = 2 * x_dot**2 + 24 * t * x

# Print the symbolic equations

print('V = ', V)
```

```
V =  24*t*x + 2*x_dot**2
```

In [103…
```
# Compute the partial derivatives

V_x = sym.diff(V, x)

V_x_dot = sym.diff(V, x_dot)

# Print the partial derivatives

print('V_x = ', V_x)

print('V_x_dot = ', V_x_dot)
```

```
V_x =  24*t
V_x_dot =  4*x_dot
```

In [104…
```
# Compute the Hessian matrix

H = sym.Matrix(sym.hessian(V, [x, x_dot]))

print('H = ', H)

if H.is_positive_semidefinite:
    print('H is Minimally Stable')
```

```
H =  Matrix([[0, 0], [0, 4]])
H is Minimally Stable
```

$$H = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix}$$

This means our solution is a minimum if the **Hessian Matrix is positive definite.**

So, this is a **minima trajectory.**

Compute Second Variation

$$\delta^2 J = \frac{1}{2} \int_{t_0}^{t_f} \{ [\, \delta x \quad \delta \dot{x} \,] \, H \begin{bmatrix} \delta x \\ \delta \dot{x} \end{bmatrix} \} dt$$

at Optimal Trajectory $x^*(t)$,

$$x^*(t) = t^3 - 1.117t$$

$$x^*(t) = t^3 + 2.687t$$

In [105…

```
def second_variation(x, H, t0, tf):
    x_dot = sym.symbols('x_dot')
    x_dot = sym.diff(x, t)

    del_x = sym.symbols('del_x')
    del_x_dot = sym.symbols('del_x_dot')

    del_x = sym.diff(x, t)
    del_x_dot = sym.diff(x_dot, t)

    m1 = sym.Matrix([[del_x, del_x_dot]])

    m2 = sym.Matrix([[del_x], [del_x_dot]])

    mat = m1 * H * m2

    ans = sym.integrate(mat[0], (t, t0, tf)) / 2

    return ans
```

Optimal Trajectory 1:

$$x^*(t) = t^3 - 1.117t$$

$$t_0 = 0$$

$$t_f = 1.55$$

In [106…

```
x = t**3 - 1.117*t
t0 = 0
tf = 1.55
second_variation(x, H, t0, tf)
```

Out[106]: 89.373

$$\delta^2 J = 89.373$$

Optimal Trajectory 2:

$$x^*(t) = t^3 + 2.687t$$

$$t_0 = 0$$

$$t_f = 0.64$$

In [107…

```
x = t**3 + 2.687*t
t0 = 0
tf = 0.64
second_variation(x, H, t0, tf)
```

Out[107]: 6.291456

$$\delta^2 J = 6.291456$$