Write our kinematic constraints and symbolic variables

```
syms F tau M m L t g x(t) theta(t)

% Cart Position is just x
disp_eq(x, x, theta, t, "Cart Position");
```

```
Cart Position:
```

x

```

% Pendulum Position is function of x and theta
x_p = x + L * sin(theta);
y_p = L * cos(theta); % theta is deviation from vertical
disp_eq(x_p, x, theta, t, "Pendulum x Position");
```

```
Pendulum x Position:
```

$x + L\sin(\theta)$

```
disp_eq(y_p, x, theta, t, "Pendulum y Position");
```

```
Pendulum y Position:
```

$L\cos(\theta)$

```

% Cart Velocity
v_x = diff(x, t);
disp_eq(v_x, x, theta, t, "Cart Velocity");
```

```
Cart Velocity:
```

$\dot{x}$

```

% Pendulum Velocity
v_px = diff(x_p, t);
v_py = diff(y_p, t);
disp_eq(v_px, x, theta, t, "Pendulum x Velocity");
```

```
Pendulum x Velocity:
```

$\dot{x} + L\,\dot{\theta}\cos(\theta)$

```
disp_eq(v_py, x, theta, t, "Pendulum y Velocity");
```

```
Pendulum y Velocity:
```

$-L\,\dot{\theta}\sin(\theta)$

Find formulas for energies

```
% Cart Kinetic Energy T
T_c = 1/2 * M * diff(x, t) ^ 2;
disp_eq(T_c, x, theta, t, "Cart KE");
```

Cart KE:

$$\frac{M\,\dot{x}^2}{2}$$

```
% Pendulum Kinetic Energy T
T_p = simplify(1/2 * m * (v_px^2 + v_py^2));
disp_eq(T_p, x, theta, t, "Pendulum KE");
```

Pendulum KE:

$$\frac{m\,\left(L^2\,\dot{\theta}^2+2\cos(\theta)\,L\,\dot{\theta}\,\dot{x}+\dot{x}^2\right)}{2}$$

```
% Total KE
T = simplify(T_c + T_p);
disp_eq(T, x, theta, t, "Total KE");
```

Total KE:

$$\frac{m\,\left(L^2\,\dot{\theta}^2+2\cos(\theta)\,L\,\dot{\theta}\,\dot{x}+\dot{x}^2\right)}{2}+\frac{M\,\dot{x}^2}{2}$$

```
% Potential energy of cart is zero
disp_eq(0, x, theta, t, "Cart PE");
```

Cart PE:

$$0$$

```
% Pendulum PE = Total PE:
V = m * g * L * cos(theta);
disp_eq(V, x, theta, t, "Pendulum PE");
```

Pendulum PE:

$$L\,g\,m\cos(\theta)$$

Calculate the largrangian and the lagrangian equations for our system

```
% Lagrangian
lagrange = simplify(T - V);
L_sym = disp_eq(lagrange, x, theta, t, "Lagrangian");
```

Lagrangian:

$$\frac{M\,\dot{x}^2}{2}+\frac{m\,\dot{x}^2}{2}+\frac{L^2\,m\,\dot{\theta}^2}{2}-L\,g\,m\cos(\theta)+L\,m\,\dot{\theta}\,\dot{x}\cos(\theta)$$

```
% External Force associated with x:
Q_x = F;
```

```
% External Torque associated with theta:
Q_theta = 0;
```

```
% Apply lagrange's equation
eq_1 = diff(diff(lagrange, diff(x, t)), t) - diff(L, x) == Q_x;
eq_2 = diff(diff(lagrange, diff(theta, t)), t) - diff(L, theta) == Q_theta;
eq_1_sym = disp_eq(eq_1, x, theta, t, "Lagrange Equation 1");
```

```
Lagrange Equation 1:
```

$$-L\,m\sin(\theta)\,\dot{\theta}^2 + M\,\ddot{x} + m\,\ddot{x} + L\,m\,\ddot{\theta}\cos(\theta) = F$$

```
eq_2_sym = disp_eq(eq_2, x, theta, t, "Lagrange Equation 2");
```

```
Lagrange Equation 2:
```

$$L^2\,m\,\ddot{\theta} + L\,m\,\ddot{x}\cos(\theta) - L\,m\,\dot{\theta}\,\dot{x}\sin(\theta) = 0$$

Solve for: $\ddot{x}$ and $\ddot{\theta}$

```
syms x_dot theta_dot x_ddot theta_ddot

[x_ddot, theta_ddot] = solve([eq_1_sym, eq_2_sym], [x_ddot, theta_ddot]);
x_ddot = simplify(x_ddot);
theta_ddot = simplify(theta_ddot);
disp("Solutions for x_ddot and theta_ddot:")
```

```
Solutions for x_ddot and theta_ddot:
```

```
disp(x_ddot)
```

$$\frac{L\,m\sin(\theta)\,\dot{\theta}^2 - m\,\dot{x}\cos(\theta)\sin(\theta)\,\dot{\theta} + F}{-m\cos(\theta)^2 + M + m}$$

```
disp(theta_ddot)
```

$$-\frac{F\cos(\theta) - M\,\dot{\theta}\,\dot{x}\sin(\theta) - m\,\dot{\theta}\,\dot{x}\sin(\theta) + L\,m\,\dot{\theta}^2\cos(\theta)\sin(\theta)}{L\,\left(-m\cos(\theta)^2 + M + m\right)}$$

```
x_sym = sym('x');
theta_sym = sym('theta');
X = [x_sym, x_dot, theta_sym, theta_dot]; % can't use the function sym
U = [F, tau]; % no torque control so jacobian second col is 0
X_dot = [x_dot, x_ddot, theta_dot, theta_ddot];

% this part is inspired by Jason's matlab code to collect coefficients of X
% in X_dot
```

```
X_dot_x = simplify(collect(X_dot, X));
Jacobi_A = jacobian(X_dot_x, X);
X_dot_u = simplify(collect(X_dot, U));
Jacobi_B = jacobian(X_dot, U);

disp("Jacobian Matrices A and B:")
```

```
Jacobian Matrices A and B:
```

```
disp(Jacobi_A)
```

$$\left(\begin{array}{llll} 0 & 1 & & \\ 0 & -\frac{m\,\dot{\theta}\cos(\theta)\sin(\theta)}{\sigma_1} & & \frac{L\,m\,\dot{\theta}^2\cos(\theta)-m\,\dot{x}\,\dot{\theta}\cos(\theta)^2+m\,\dot{x}\,\dot{\theta}\sin(\theta)^2}{\sigma_1}-\frac{2\,m}{} \\ 0 & 0 & & \\ 0 & \frac{M\,\dot{\theta}\sin(\theta)+m\,\dot{\theta}\sin(\theta)}{L\,\sigma_1} & \frac{F\sin(\theta)-L\,m\,\dot{\theta}^2\cos(\theta)^2+L\,m\,\dot{\theta}^2\sin(\theta)^2+M\,\dot{\theta}\,\dot{x}\cos(\theta)+m\,\dot{\theta}\,\dot{x}\cos(\theta)}{L\,\sigma_1}+ & \end{array}\right.$$

where

$$\sigma_1 = -m\cos(\theta)^2 + M + m$$

```
disp(Jacobi_B)
```

$$\begin{pmatrix} 0 & 0 \\ \frac{1}{-m\cos(\theta)^2+M+m} & 0 \\ 0 & 0 \\ -\frac{\cos(\theta)}{L\,(-m\cos(\theta)^2+M+m)} & 0 \end{pmatrix}$$

Test controllability conditions and see LQR matrices

```
% say cart mass is 5kg, pendulum mass is 0.5kg, L = 1 m, g = 9.81 m/s^2
A = subs(Jacobi_A, [M, m, L, g, F], [5, 0.5, 1, 9.81, 0]);
B = subs(Jacobi_B, [M, m, L, g], [5, 0.5, 1, 9.81]);
fprintf("A and B Matrices: \n")
```

```
A and B Matrices:
```

```
disp(A)
```

$$\begin{pmatrix} 0 & 1 & 0 & \\ 0 & \frac{\sigma_2}{2\,\sigma_1} & -\frac{\frac{\dot{\theta}^2\cos(\theta)}{2}-\frac{\dot{x}\,\dot{\theta}\cos(\theta)^2}{2}+\frac{\dot{x}\,\dot{\theta}\sin(\theta)^2}{2}}{\sigma_1}-\frac{\cos(\theta)\sin(\theta)\left(\frac{\dot{\theta}^2\sin(\theta)}{2}-\frac{\dot{\theta}\,\dot{x}\cos(\theta)\sin(\theta}{2}\right.}{\sigma_1{}^2} \\ 0 & 0 & 0 & \\ 0 & -\frac{11\,\dot{\theta}\sin(\theta)}{2\,\sigma_1} & -\frac{-\frac{\dot{\theta}^2\cos(\theta)^2}{2}+\frac{\dot{\theta}^2\sin(\theta)^2}{2}+\frac{11\,\dot{x}\,\dot{\theta}\cos(\theta)}{2}}{\sigma_1}-\frac{\cos(\theta)\sin(\theta)\left(\frac{11\,\dot{\theta}\,\dot{x}\sin(\theta)}{2}-\frac{\dot{\theta}^2\cos(\theta)\,\mathrm{si}}{2}\right.}{\sigma_1{}^2} \end{pmatrix}$$

where

$$\sigma_1 = \frac{\cos(\theta)^2}{2}-\frac{11}{2}$$

$$\sigma_2 = \dot{\theta}\cos(\theta)\sin(\theta)$$

```
disp(B)
```

$$\begin{pmatrix} 0 & 0 \\ -\frac{1}{\frac{\cos(\theta)^2}{2}-\frac{11}{2}} & 0 \\ 0 & 0 \\ \frac{\cos(\theta)}{\frac{\cos(\theta)^2}{2}-\frac{11}{2}} & 0 \end{pmatrix}$$

```
% system is not full rank if jacobian is evaluated at unstable equilibrium point
% as an experiment, I evaluate it with slight angular velocity and
% theta=pi/4
theta_0 = pi/4;
theta_dot_0 = 0.001
```

```
theta_dot_0 = 1.0000e-03
```

```
A_0 = double(subs(A, [x_sym, x_dot, theta_sym, theta_dot], [0, 0, theta_0,
theta_dot_0]))
```

```
A_0 = 4×4
         0    1.0000         0         0
         0   -0.0000    0.0000    0.0001
         0         0         0    1.0000
         0    0.0007    0.0000   -0.0001
```

```
B_0 = double(subs(B, [x_sym, x_dot, theta_sym, theta_dot], [0, 0, theta_0,
theta_dot_0]))
```

```
B_0 = 4×2
         0         0
    0.1905         0
         0         0
   -0.1347         0
```

```
rank(ctrb(A_0,B_0))
```

```
ans = 4
```

```
% cost matrices
Q = diag([10, 0, 10, 0]);
R = diag([0.1, 0.1]);

% lqr gain
[K, S, e] = lqr(A_0, B_0, Q, R);
disp(K)
```

```
   1.0e+04 *

   -0.0010    3.2240    0.0014    4.5578
    0.0000   -0.0000   -0.0000   -0.0000
```

```
function equation_s = disp_eq(equation, x, theta, t, name)
    % replace second derivatives
    equation_s = subs(equation, diff(x, t, 2), sym('x_ddot'));
    equation_s = subs(equation_s, diff(theta, t, 2), sym('theta_ddot'));

    % replace first derivatives
    equation_s = subs(equation_s, diff(x, t), sym('x_dot'));
    equation_s = subs(equation_s, diff(theta, t), sym('theta_dot'));

    % replace function terms
    equation_s = subs(equation_s, x, sym('x'));
    equation_s = subs(equation_s, theta, sym('theta'));

    % display equation
    fprintf(name + ":\n")
    disp(equation_s)
    fprintf("\n")
end
```