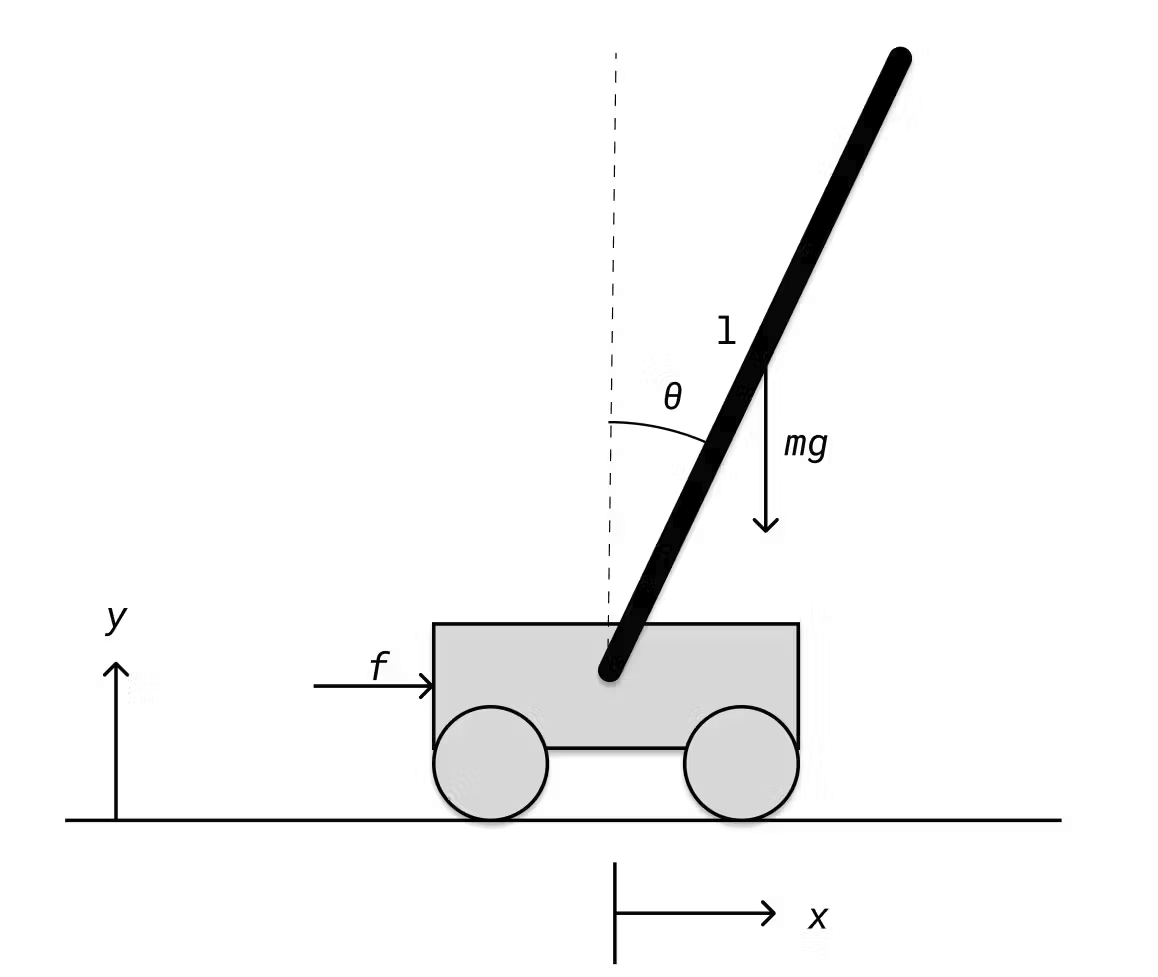
# 1 Find a program to simulate Cart Pole

The Cart-Pole system is what's known as a nonlinear dynamic system.



Here are the code of dynamic system

%% Set Parameters

m=0.375; %Pole Mass (kg)

l=0.1; %Pole Length (m)

M=07.0; %Cart Mass (kg)

g=9.81; %Gravity (m/s^2)

%% Non-Linear Dynamics

dudt=@(t,u)[u(2);...

(-m\*l\*sin(u(3))\*u(4)^2+m\*g\*cos(u(3))\*sin(u(3)))...

/(M+m-m\*cos(u(3))^2);...

u(4);...

(-m\*l\*cos(u(3))\*sin(u(3))\*u(4)^2+M\*g\*sin(u(3)))...

/(l\*(M+m-m\*cos(u(3)))^2)];

%% Run Simulation

u0=[0 0 pi/4 0]; %Initial Conditions

tspan = 0:0.005:5; %Simulation Time 0:dt:TMax

[t,u] = ode45(dudt,tspan,u0); %Compute Simulation

%% Animate

Animate

%% Plot system response

PlotResponse

I will explain this nonlinear dynamics equation below

%% Non-Linear Dynamics

dudt=@(t,u)[u(2);...

(-m\*l\*sin(u(3))\*u(4)^2+m\*g\*cos(u(3))\*sin(u(3)))...

/(M+m-m\*cos(u(3))^2);...

u(4);...

(-m\*l\*cos(u(3))\*sin(u(3))\*u(4)^2+M\*g\*sin(u(3)))...

/(l\*(M+m-m\*cos(u(3)))^2)];

This code describes the nonlinear dynamical equations of the inverted pendulum (cartpole) system, corresponding to the following physical equations:

The change in position of the cart (u(2)):

This component represents the velocity of the cart, so it is equal to the velocity of the cart itself, i.e. "u(2)".

Change in velocity of the cart:

This component represents the change in the velocity of the cart, which is affected by two factors:

The gravitational potential energy of the pole : " -m \* l \* sin(u(3)) \* u(4)^2 ", where "m" is the mass of the pole, "l" is the length of the pole, "u(3)" is the angle of the pole, and "u(4)" is the angular velocity of the pole. This term represents the force exerted by the pole on the cart downward due to gravity and the component due to the angle of the pole "u(3)".

Kinetic energy of the pole: " m \* g \* cos(u(3)) \* sin(u(3))", where "m" is the mass of the pole, "g" is the acceleration due to gravity, and "u(3)" is the angle of the pole. This term represents the effect of the kinetic energy of the pole, due to its angular velocity "u(4)", on the velocity of the cart.

The difference between these two terms results in a change in the velocity of the cart.

The change in angular velocity of the pole (u(4)):

This component indicates that the angular velocity of the pole is constant and therefore it is equal to the angular velocity of the pole itself, i.e. "u(4)".

Change in angular velocity of the pole:

This component represents the change in the angular velocity of the pole, which is affected by two factors:

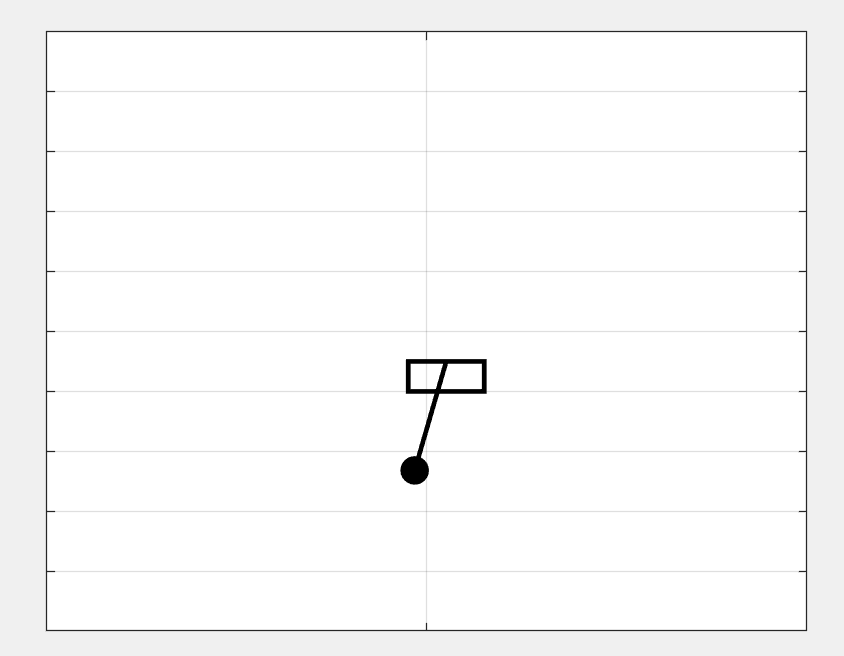
The angular acceleration of the pole: " -m \* l \* cos(u(3)) \* sin(u(3)) \* u(4)^2 ", where "m" is the mass of the pole, "l" is the length of the pole, "u(3)" is the angle of the pole, and "u(4)" is the angular velocity of the pole. This term represents the angular acceleration of the pole due to its angular velocity "u(4)".

The gravitational torque of the pole: " M \* g \* sin(u(3))", where "M" is the mass of the cart, "g" is the gravitational acceleration, and "u(3)" is the angle of the pole. This term represents the torque due to the angle of the pole "u(3)".

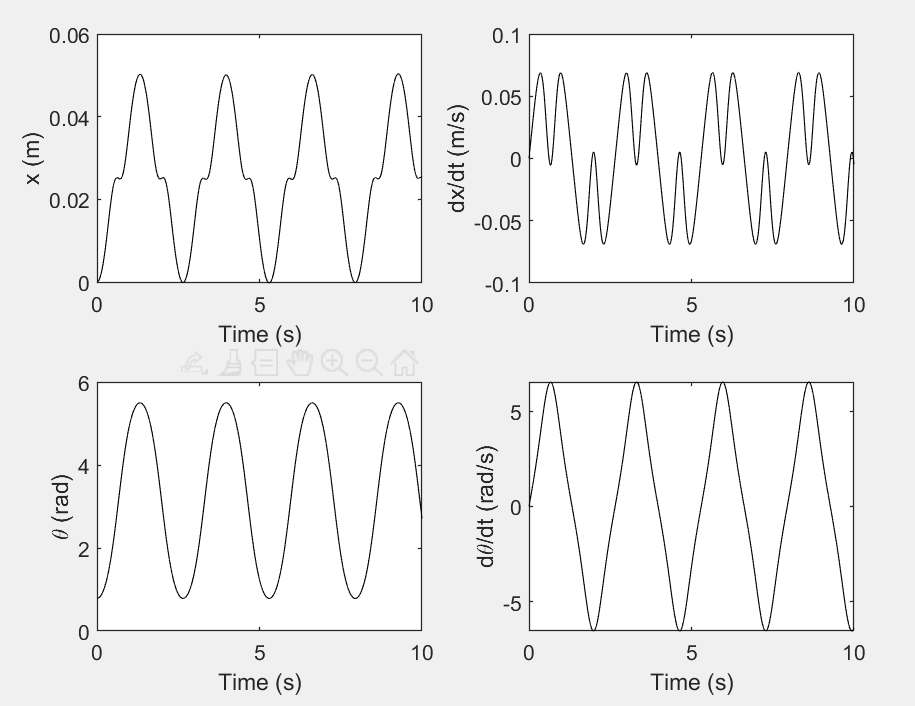
The difference between these two terms results in a change in the angular velocity of the pole.

# 2 Change the mass of cart or pole, and explain the differences behaviours

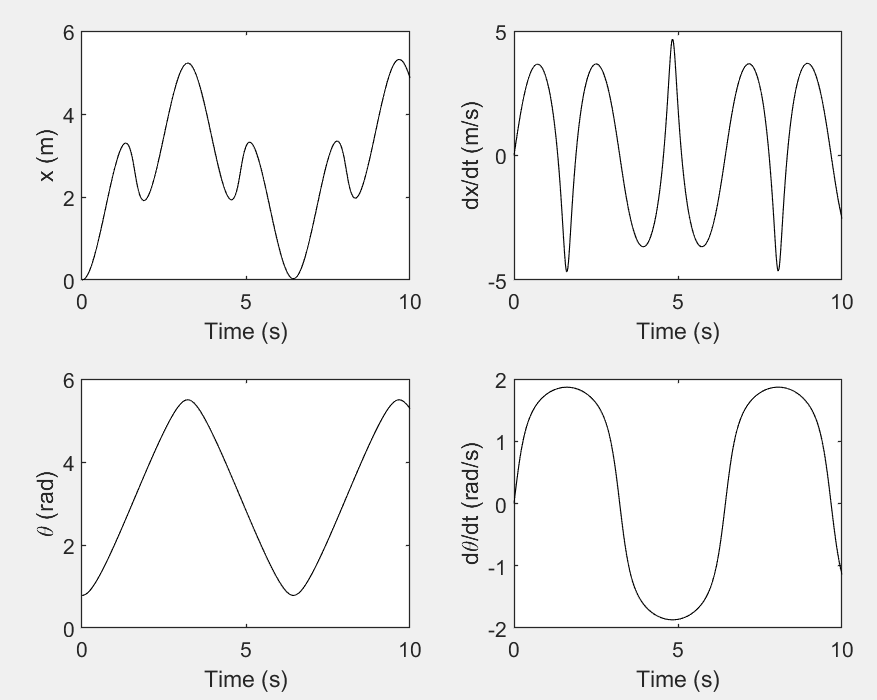
The program runs with the following results:



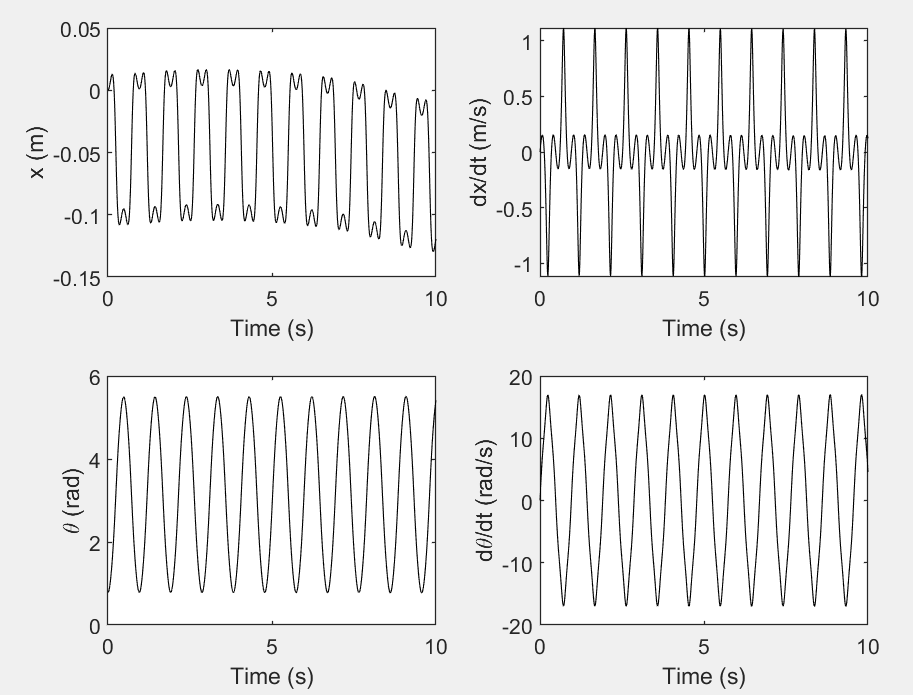
this is a visualisation of the trolley and the bar



This is the velocity and acceleration curves when the mass of the pole is 0.375kg and the mass of the cart is 7Kg.



Here are the velocity and acceleration curves for a mass of 7kg for the pole and 0.375kg for the car.



Here are the velocity and acceleration curves for a mass of 0.375kg for the pole and 0.375kg for the car.(car and pole has the same weight)

It can be seen that when the mass of the pole and the mass of the car are similar, the two are conserved in momentum in the horizontal direction and the velocity exchange frequency is fast. When the mass of the pole is greater than the mass of the car, momentum is conserved in the horizontal direction, and the displacement and velocity of the car will be large, reflecting the greater influence of the momentum of the pole on the system. When the mass of the pole is less than the mass of the car, the displacement and velocity of the car will be small. In both cases velocities are not exchanged as often as when the masses are similar.

# 3 Find a program to simulate LQR controller, and make it work when the pole has a heavier mass

Here are the code of LQR control

clear all;

clc;

%% Set Parameters

global m,

global l;

global M;

global g;

m=5; %Pole Mass (kg)

l=0.1; %Pole Length (m)

M=0.3; %Cart Mass (kg)

g=9.81; %Gravity (m/s^2)

%% Run Simulation

x0=[-0.25 0 -0.7 0]; %Initial Conditions

tspan = 0:0.005:10; %Simulation Time 0:dt:TMax

[t,u] = ode45(@odefunc,tspan,x0); %Compute Simulation

%% Animate

Animate

%% Plot system response

PlotResponse

%%%%% ODE Function %%%%%

function dxdt=odefunc (t,x)

global m,

global l;

global M;

global g;

%% Design stabilizing LQR controller

%Linearize Dynamics about swing up position u=[0 0 0 0]. System is of the

%form xdot=Ax+Bu.

A=[0 1 0 0;...

0 0 m\*g/M 0;...

0 0 0 1;...

0 0 (m\*g+M\*g)/(M\*l) 0];

B=[0 1/M 0 1/(M\*l)]';

%Define LQR parameters

StateCost=10; %Cost of state

ControlCost=0.01; %Cost of control

Q=[StateCost 0 0 0; 0 0 0 0; 0 0 StateCost 0; 0 0 0 0]; %State cost matrix

R=ControlCost; %Control cost

K=lqr(A,B,Q,R); %Calculate LQR gain

x;

%% Non-linear dynamics with LQR controller

dxdt=[x(2);...

(-m\*l\*sin(x(3))\*x(4)^2+(-K\*x)+m\*g\*cos(x(3))\*sin(x(3)))...

/(M+m-m\*cos(x(3))^2);...

x(4);...

(-m\*l\*cos(x(3))\*sin(x(3))\*x(4)^2+(-K\*x)\*cos(x(3))+m\*g\*sin(x(3))+M\*g\*sin(x(3)))...

/(l\*(M+m-m\*cos(x(3)))^2)];

End

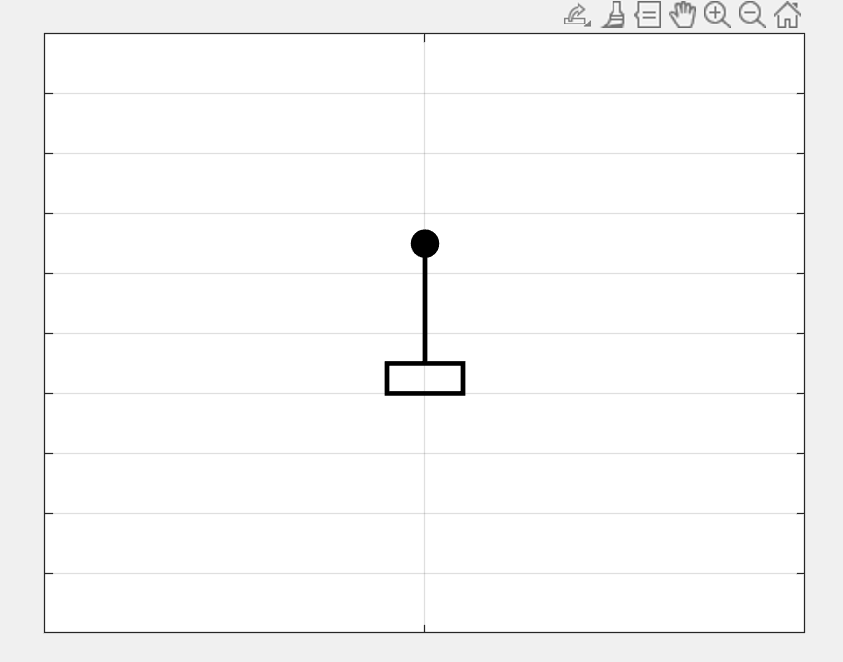
This part of the code defines the design of the LQR controller, including the linearisation of the system. It computes the state space matrix A and the input matrix B which are used for the design of the LQR controller.

The parameters of the LQR controller are also defined, including the weighting matrices Q and R for the state and control costs, and the calculation of the LQR control gain matrix K using the lqr function.

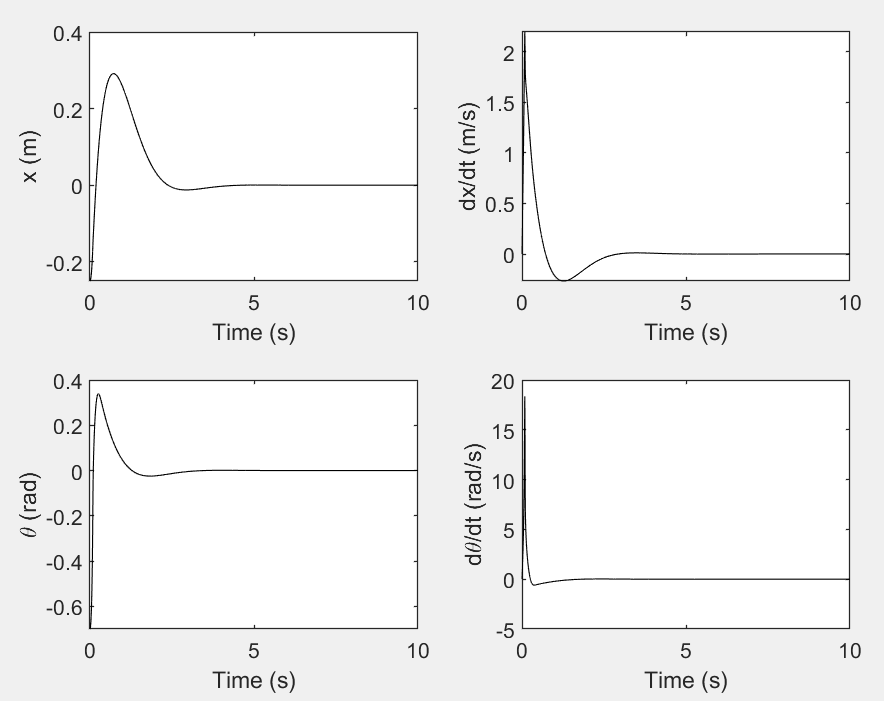
The last part calculates the nonlinear dynamics equations of the inverted pendulum system and applies the LQR controller. The equations of dynamics are the derivatives dxdt of the state vector x, including changes in the position, velocity, rod angle and angular velocity of the cart. These equations describe the evolution of the system in a given state, taking into account the effect of the control inputs (-K\*x) on the system.

I'm not going to delve into the principles of LQR as I'm not a student of control studies.

The program runs with the following results:



Here is the final stable state of it



We can see that it can stabilize such a diversity of systems!