# Introduction

Before performing numerical evaluations at critical stages in this coursework, you should substitute the following parameter values into your derivation:

# 1. Linearise the non-linear differential equation

In this task, I am finding the unstable equilibrium point and linearise the equation around this point. An equilibrium point is any point that makes all rates 0. Balancing the torques around the centre of mass of the pendulum leads to the following equation of motion:



CONVERTING THE ABOVE EQUATION TO THE ONE BELOW USING JACOBIAN MATRICES!!!!!!!



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LINEARISATION OF THE NON-LINEAR DIFFERENTIAL EQUATION.



LET:





NOW:



Rearrange in terms of the highest derivative:



Choosing the state variables:







The equilibrium of a non-inverted pendulum occurs when:



The equilibrium of an inverted pendulum occurs when:



Rewritting our equations as functions.





The Jacobian matrices for system and control are given by:

Differentiating w.r.t. state variables:



Differentiating w.r.t. to control input:







Substituting the conditions:

State variables:







Control Input:







The state space equation for the pendulum is therefore:



Substituting the real value of G:



The linear solution of the pendulum:







Substituting the original values:









 [LINEAR SOLUTION]

Substituting the values:

Substituting the conditions when :

State variables:







Control Input:







The matrix expression for the pendulum when is therefore:



Substituting the real value of G:



# 2. Write down the state space model of the system



:

 [OUTPUT EQUATION]  



Rearrange the equation in terms of the highest derivative:



**LET:**





We have a derivative at the RHS [] that we do not want and we want to get rid of it by adding the term both sides of the equation.



Choosing the state variables again because we made changes to the equation:





Earlier, we said ;



 [output equation]

 [1st state equation]

 [2nd state equation]

First, we substitute the values of  and into the 2nd state equation above.









 \*\*\*\*\*



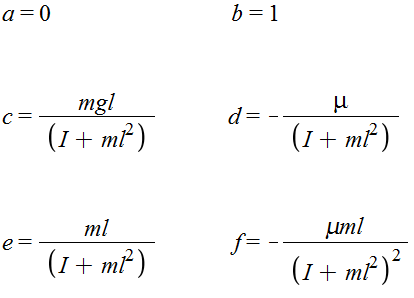


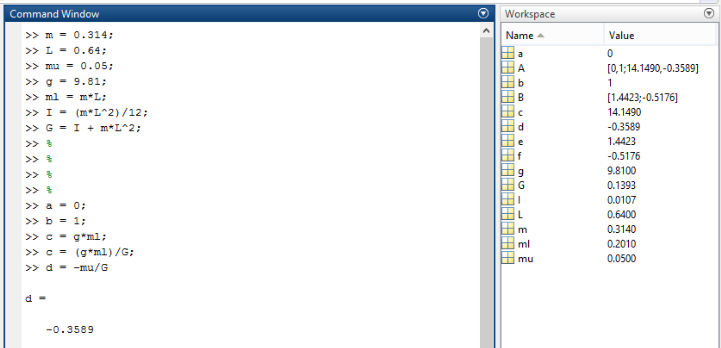


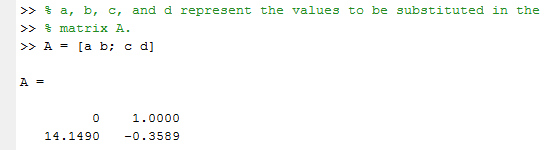
# 3. Observability, controllability and stability

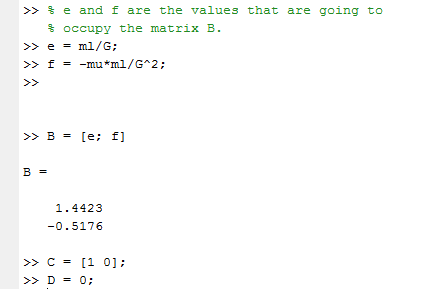
After linearisation of the non-linear ODE using the Jacobian Matrices, we then calculated the state space equation of the system and we got the following solution: 

In-order to make things easier in Matlab, I renamed and simplified the matrices above and this can be shown by the image below:



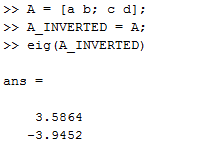






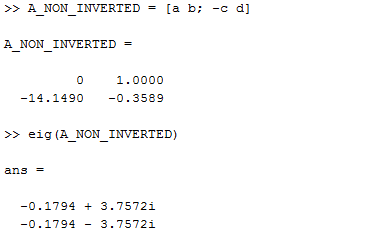
**STABILITY**: The stability of the equilibrium is examined by calculating the eigenvalues λ of the system matrix A.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



The solution to our equation includes positive eigenvalues therefore the system is unstable.

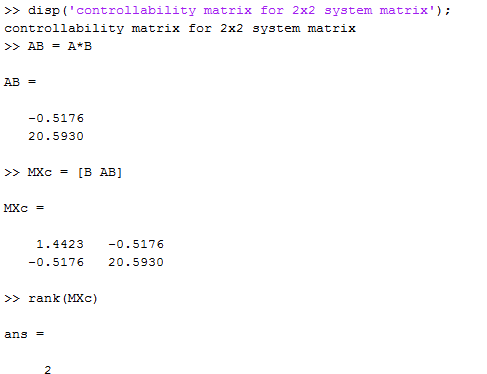
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Eigenvalues almost on imaginary axis therefore stable (would be marginally stable without damping term).

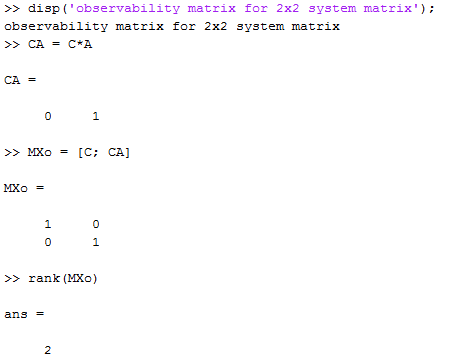
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CONTROLLABILITY**:



This is a full rank matrix so the system is controllable.

**OBSERVABILITY**:



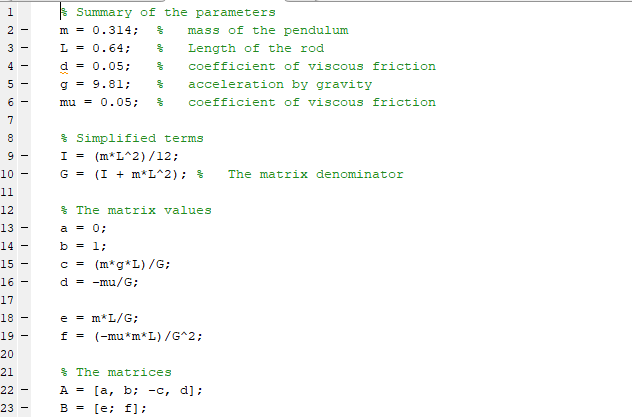
This is a full rank matrix so the system is observable.

# 4. Simulate your state space module using the Matlab ode45 function

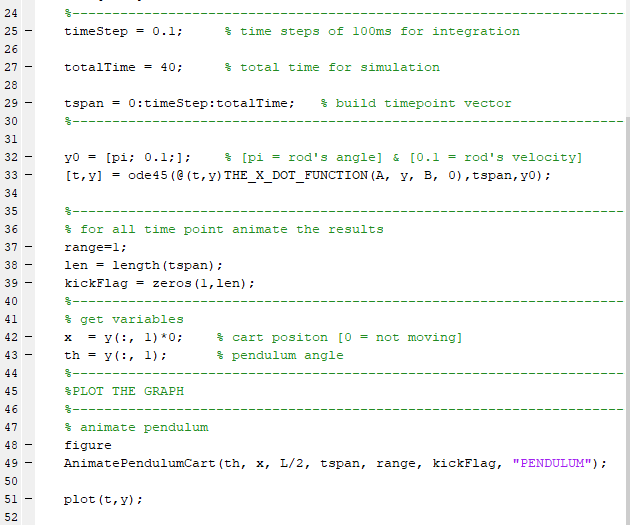


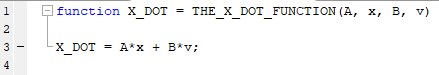
Simulate your state space module using the Matlab ode45 function:

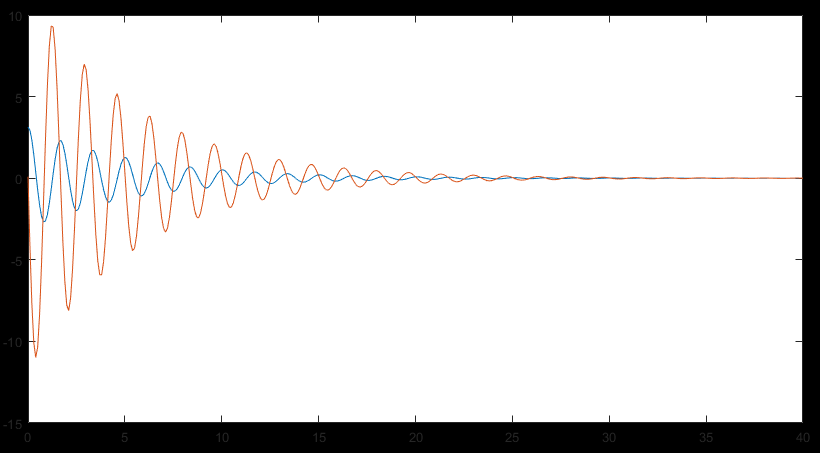
In Matlab, I created a file that contains all the parameters of my state space equation written above.



We were provided with a function that simulates the pendulum in an animation and we are calling this function after we solved the 1st differential equation using the ODE45.







# 5. Design a state feedback controller

Design a state feedback controller

First write down the matrix equation for state feedback control, indicating the form of the gain vector:

# 6. Implement the controller/observer system using Euler integration

Implement the controller/observer system using Euler integration

# 7. Add a Luenberger observer to your state feedback controller

# 8. Augment positional state into your state space model

Previously, we noticed that the

# 9. Implement the augmented state feedback controller

# 9. Implement the augmented state feedback controller

# 10. Run the inverted pendulum demos