Linear Control Analysis of Double Link Robotic Arm

Siddharth Rajagopal
Department of Mechanical Engineering
University of Washington,
Seattle, Washington, USA

I. INTRODUCTION

Robotic Arms are widely used in the industry for solving various demanding applications. The flexibility that just by changing the end tool of the robotic arm, it can be used to solve difficult and repetitive tasks like welding, picking and placing, bolting etc. makes at very practical.

For ease of study, a robotic arm that has two links is studied that has movement in the 2-D plane. The dynamics of the robotic arm is used to derive the non-linear equations of motion, which is later linearized around an equilibrium point. The controllability and observability of this system is obtained and the control is used to drive the system to stability in a finite time. Later the states are estimated from the output signals obtained and a feedback law is applied to the system.

II. MODEL

A double pendulum model was used to study the equations of motion of the 2 link robotic arm using Euler-Lagrange method.

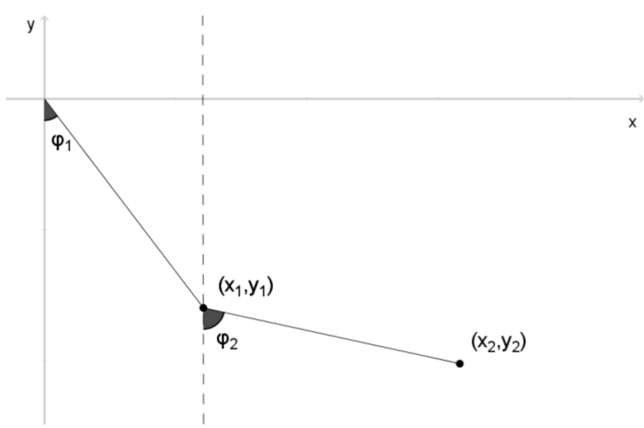


fig 1. A mathematical double pendulum

We have the relations :

$$
\begin{aligned}
x_1 &= \sin\varphi_1 \\
y_1 &= \cos\varphi_1 \\
x_2 &= \sin\varphi_1 + \sin\varphi_2 \\
y_2 &= \cos\varphi_1 + \cos\varphi_2
\end{aligned}
$$

A. States of the System

The state of the system at any point of time can be described using the following state variables:

- Angle subtended by each arm: φ_1, φ_2
- Angular velocity of each arm: ω_1, ω_2

B. Equation of motion:

$$
\begin{aligned}
\dot{\varphi}_1 &= \omega_1 \\
\dot{\varphi}_2 &= \omega_2 \\
\dot{\omega}_1 &= \frac{-3\sin\varphi_1 - \sin(\varphi_1 - 2\varphi_2) - 2\sin(\varphi_1-\varphi_2)(\omega_2^2 + \omega_1^2\cos(\varphi_1-\varphi_2))}{3-\cos(2\varphi_1-2\varphi_2)} \\
\dot{\omega}_2 &= \frac{2\sin(\varphi_1-\varphi_2)(2\omega_1^2 + 2\cos\varphi_1 + \omega_2^2\cos(\varphi_1-\varphi_2))}{3-\cos(2\varphi_1-2\varphi_2)}.
\end{aligned}
$$

C. Inputs:

The inputs to the system are the angular acceleration U_1 and U_2 at each of the arms.

D. Outputs:

The outputs of the system are all the states of the system namely : $\varphi_1, \varphi_2, \omega_1, \omega_2$

E. Assumptions:

The mathematical double pendulum considered is assumed to have no friction, with massless pendulum links and motion occurring in a 2D plane with no gravity. The length of each link is assumed to be one unit. The angles are measured with respect to negative y axis.

III. ANALYSIS

A. Equilibrium:

The equilibrium points for the double pendulum occur when there is no movement in the links. The double pendulum has 4 equilibrium points. When $\varphi_1, \varphi_2, \omega_1, \omega_2$ =

$$(0,0,0,0), \quad (0,\pi,0,0), \quad (\pi,0,0,0), \quad (\pi,\pi,0,0)$$

These equilibrium states are obtained when:

$$\frac{d}{dt}\begin{pmatrix}\varphi_1(t)\\ \varphi_2(t)\\ \omega_1(t)\\ \omega_2(t)\end{pmatrix} = 0$$

B. Linearisation:

The state representation of the system is given by:

$$x' = Ax + Bu$$

$$y = Cx + Du$$

The Jacobian matrix is formed at (0,0,0,0) to give the new A matrix of the form x'=Ax

$$\frac{d}{dt}\begin{pmatrix}\varphi_1(t)\\ \varphi_2(t)\\ \omega_1(t)\\ \omega_2(t)\end{pmatrix} = \underbrace{\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -2 & 1 & 0 & 0\\ 2 & -2 & 0 & 0\end{pmatrix}}_{=:A_{(0,0,0,0)}}\begin{pmatrix}\varphi_1(t)\\ \varphi_2(t)\\ \omega_1(t)\\ \omega_2(t)\end{pmatrix}$$

The linearised matrices at other stable points are :

$$A_{(\pi,0,0,0)} := \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 2 & -1 & 0 & 0\\ 2 & -2 & 0 & 0\end{pmatrix}$$

$$A_{(0,\pi,0,0)} := \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ -2 & 1 & 0 & 0\\ -2 & 2 & 0 & 0\end{pmatrix}$$

$$A_{(\pi,\pi,0,0)} := \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 2 & -1 & 0 & 0\\ -2 & 2 & 0 & 0\end{pmatrix}$$

C. Stability:

The eigen values obtained at each of these equilibrium points are:

- at $A_{(0,0,0,0)} = \pm\sqrt{2-\sqrt{2}}\cdot i, \pm\sqrt{2+\sqrt{2}}\cdot i$

is stable but not asymptotically stable.

- at $A_{(\pi,0,0,0)} = \pm 2^{\frac{1}{4}}, \pm 2^{\frac{1}{4}}\cdot i$

the zero solution is unstable.

- at $A_{(0,\pi,0,0)} = \pm 2^{\frac{1}{4}}, \pm 2^{\frac{1}{4}}\cdot i$

the zero solution is unstable.

- at $A_{(\pi,\pi,0,0)} = \pm\sqrt{2+\sqrt{2}}, \pm\sqrt{2-\sqrt{2}}$

the zero solution is unstable.

We select the linearization around the point $(\pi,\pi,0,0)$ for further analysis. The presence of saddle point indicates the unstability.

$$A_{(\pi,\pi,0,0)} := \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 2 & -1 & 0 & 0\\ -2 & 2 & 0 & 0\end{pmatrix}$$

$$B = \begin{pmatrix}0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & 1\end{pmatrix}$$

$$C = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

$$D = \begin{pmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{pmatrix}$$

D. Controllability and Observability

The controllability of the system can be easily computed by using above A, B matrices due to the time invariant nature of the system. The controllability matrix is :

$$\mathcal{C} = [B \mid AB \mid A^2B \mid A^3B] = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 & 2 & -1\\ 0 & 0 & 0 & 1 & 0 & 0 & -2 & 2\\ 1 & 0 & 0 & 0 & 2 & -1 & 0 & 0\\ 0 & 1 & 0 & 0 & -2 & 2 & 0 & 0\end{bmatrix}$$

$\mathrm{rank}(\mathcal{C}) = 4,$

The observability can be analysed from the A and C matrices which is given by :

$$\mathcal{O} = \begin{bmatrix}C\\ CA\\ CA^2\\ CA^3\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 2 & -1 & 0 & 0\\ -2 & 2 & 0 & 0\\ 2 & -1 & 0 & 0\\ -2 & 2 & 0 & 0\\ 0 & 0 & 2 & -1\\ 0 & 0 & -2 & 2\\ 0 & 0 & 2 & -1\\ 0 & 0 & -2 & 2\\ 6 & -4 & 0 & 0\\ -8 & 6 & 0 & 0\end{bmatrix}$$

$\mathrm{rank}(\mathcal{O}) = 4,$

IV. Computation

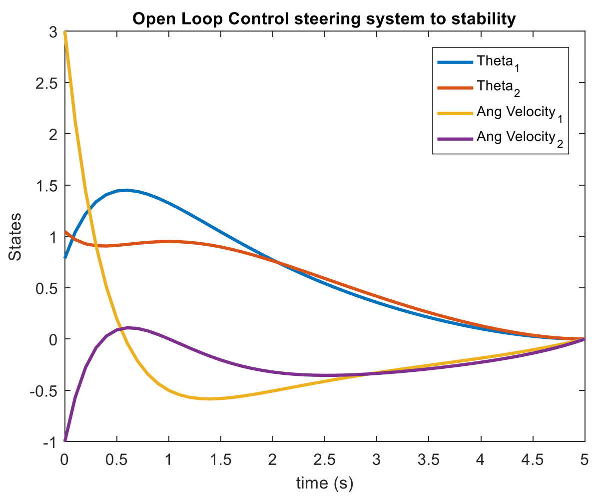
Now that we have seen that the system is both observable and controllable, we can proceed using the A, B, C and D matrices as they are above for further computation.

A. Open Loop Control

The system is to be driven from a non equilibrium state to stability. For this an initial state and a final state are decided. Using the Singular Value Decomposition (SVD) method to find the inverse of the controllability matrix will help find an input that will drive the system to the desired state.

$$u(t) = V\,\Sigma^{+} U^{T} (X_f - A_d{}^{t} X_0)$$

- $V\,\Sigma^{+} U^{T}$: decomposed parts of the controllability matrix
- X_f : Final State (at equilibrium)
- X_0 : Initial State (Unstable)



As shown above the states which are initially at unstable values are driven to an equilibrium (0,0,0,0).

B. Closed Loop Control

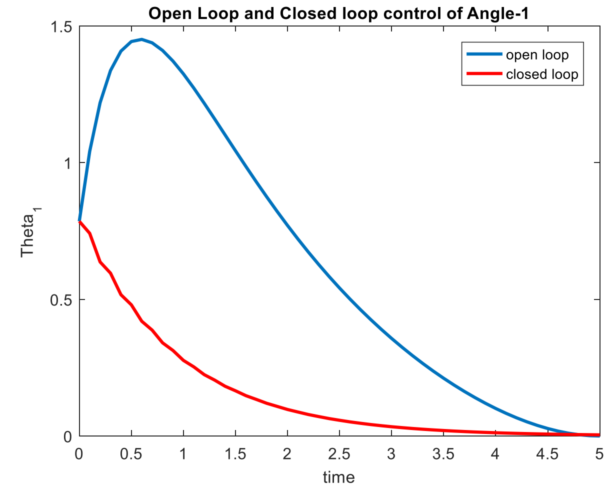
A closed loop controller uses a feedback mechanism to continuously change the input *u* to asymptotically stabilize the system given by:

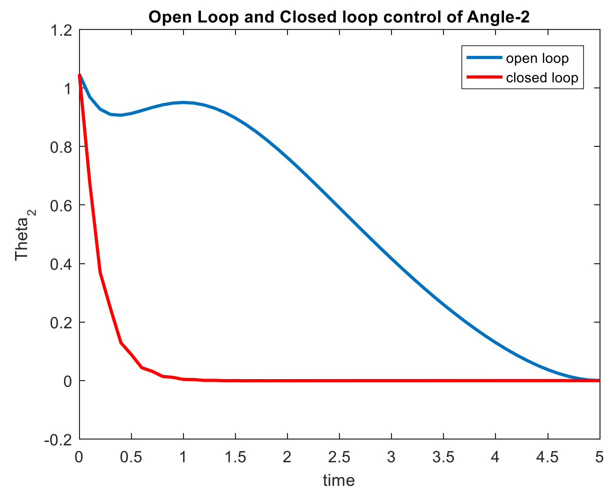
$$u = -Kx$$

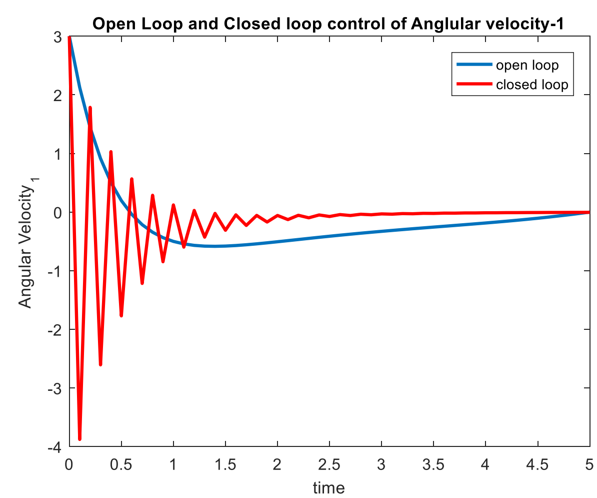
$$X' = (A - BK)\,X$$

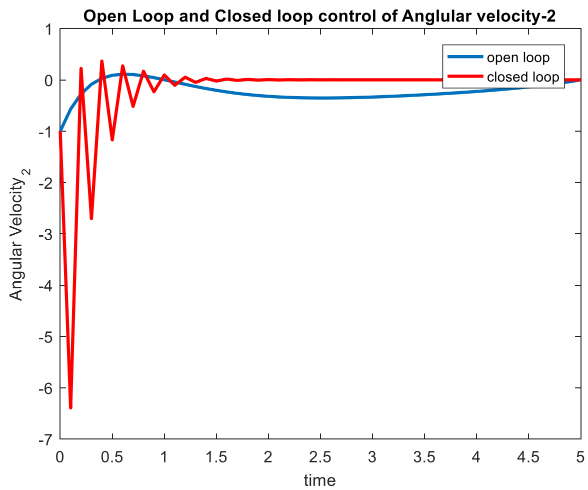
We assign a desired eigen value to yield a K matrix that solves the Pole placement problem by shifting the eigen values of the
$(A - BK)$ matrix (which represents the new system) into stability.

The following plots show the closed loop performance of the system states against their open loop performance.









C. State Estimation

The controller above was designed under the assumption that we can observe all the states. In reality the controller may be able to measure only a subset of the states. In case of a robotic arm, the angular positions may be easy to measure but measuring the velocity is comparatively difficult. So we set up linear equations relating the initial state to the input and the observed states.

$$X(0) = V_o \, \Sigma_o{}^{+} U_o{}^{T} Y$$

Here *X(0)* is the estimated initial state from the decomposed Observability matrix using Singular Value Decomposition Method and by using the Y matrix. This method helps us to estimate the state at each time step using simulation.

D. Asymptotic State Estimator

During the estimation of the initial states from the observability matrix, an error is developed due to the estimation process.

An asymptotic state estimator is used to reduce this error generated during estimation

$$e = X - \hat{X}$$

$$\dot{e} = (A - LC)e$$

$$\dot{\hat{X}} = A\hat{X} + Bu - L(\hat{Y} - Y)$$

where,

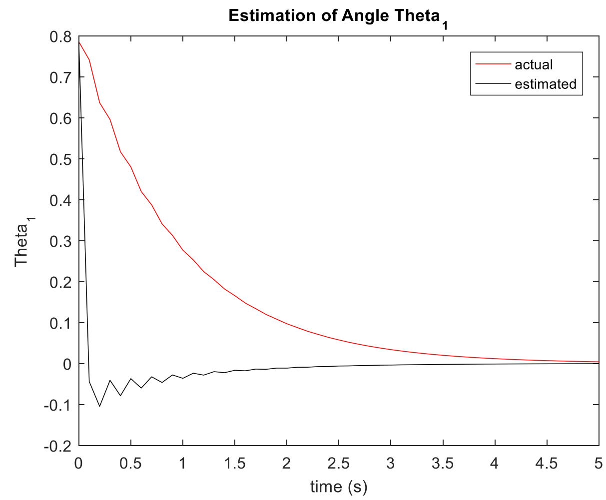
e is the error generated during the estimation process

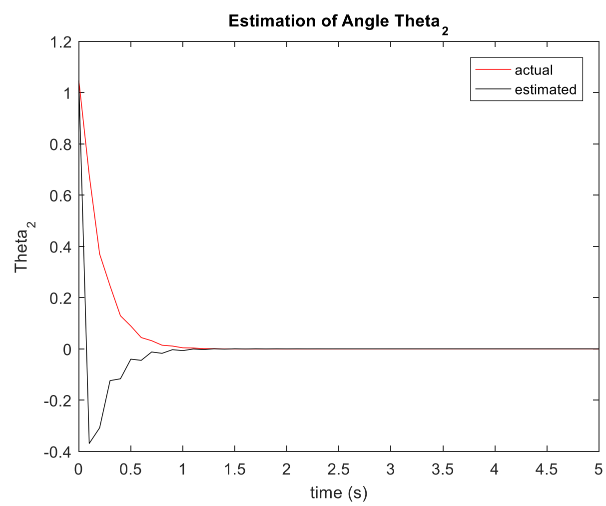
$\hat{X}$ is the estimated state

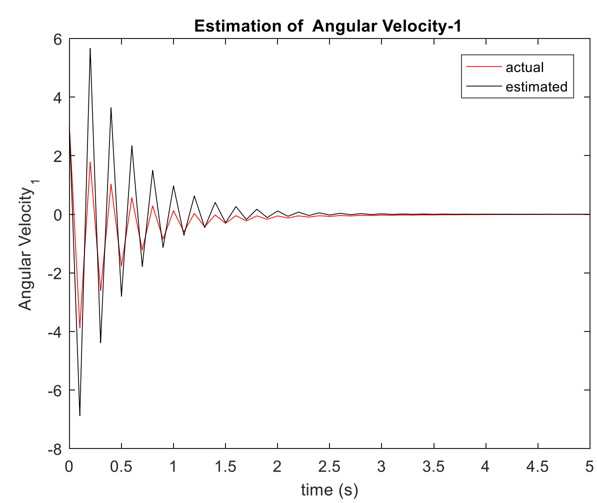
$\hat{Y}$ is the estimated output

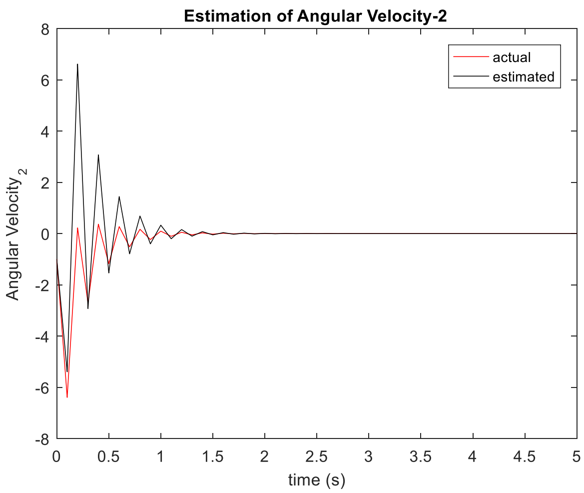
L is the pole placement matrix to stabilize the process

We can observe from the following plots that the *L* operator eventually removes the error making the actual and estimated states coincide.









E. *Closed Loop Estimator/Controller*

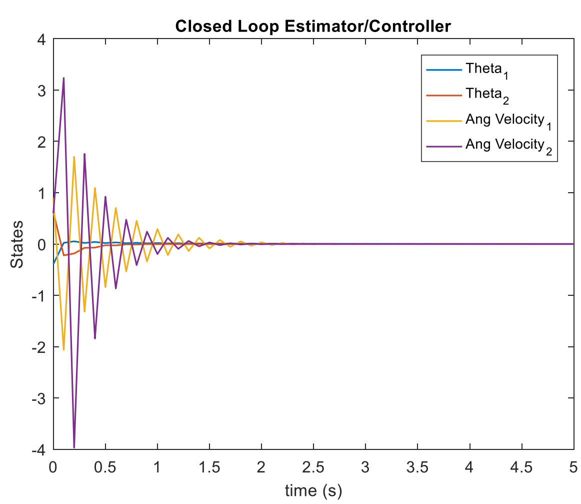
The technique of stabilising the error between the estimated and the observed states is implemented on the closed loop control method that was done in section *IV.B* to make a closed loop controller/estimator.

$$\dot{X} = (A - BK)\,\tilde{X} + BKe$$

$$\dot{e} = (A - LC)e$$

This controller/ estimator now can bring the states to equilibrium while stabilising the errors to give a coherent state.

The plot below shows the states of Theta_1, Theta_2, Angular Velocity_1 and Angular Velocity_2 of the Robotic Arm using this Controller/Estimator.

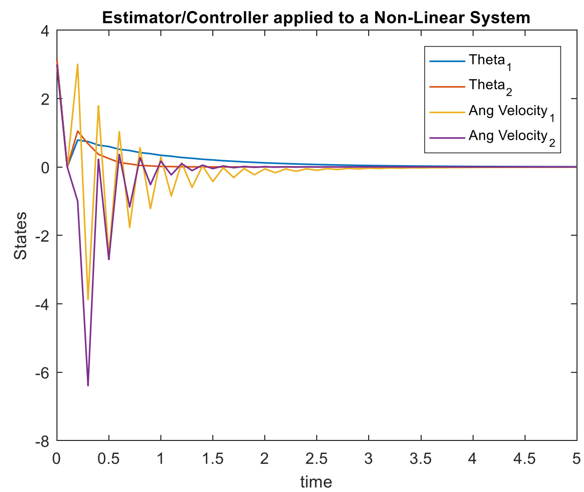


V. Application

A. *Non Linear System*

The process above were designed for a linear system linearized about a stable point. The controller/estimator that is made is now implemented on the Non-Linear system to observe the results.

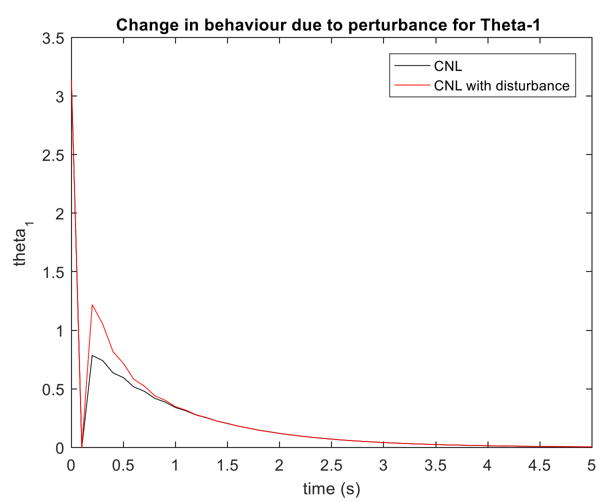
The estimated states are now processed though the non-linear set-up. From the plots below, we observe that the controller/estimator that has been developed works for the non-linear system as well

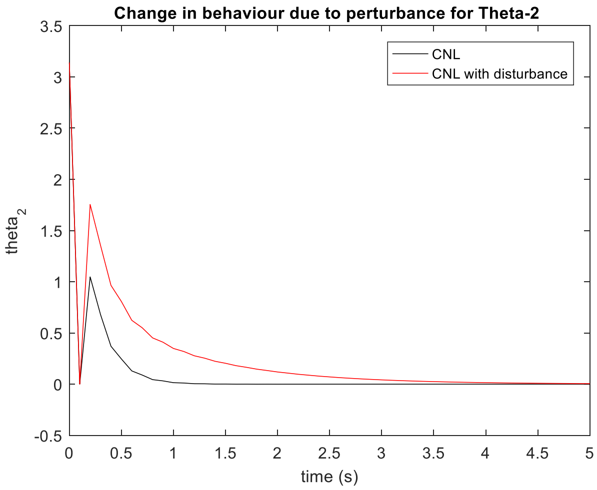


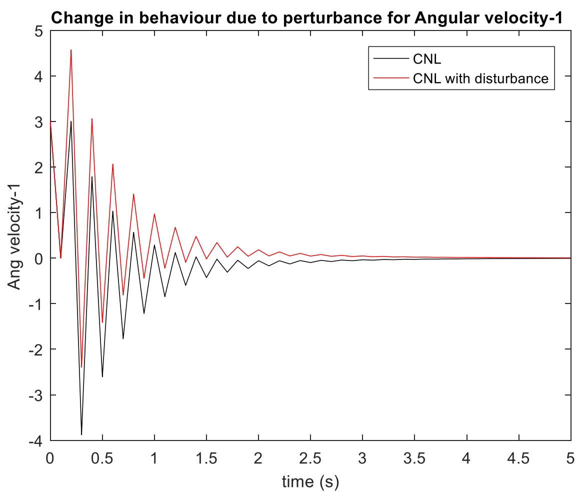
B. *Non Linear System with Perturbation*

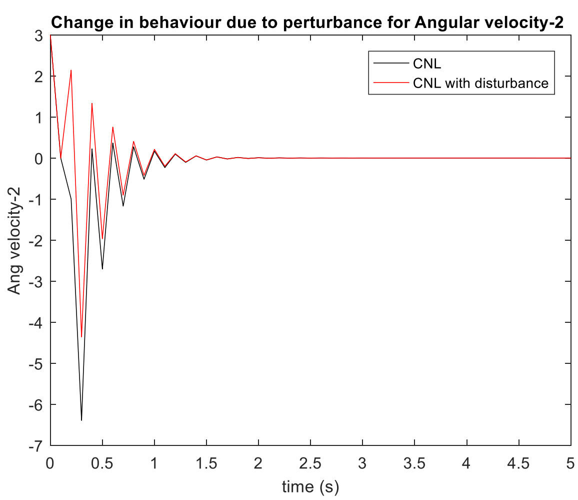
In practical applications, there will be perturbations and noises present in the signals obtained from the sensors and in the actuation process. To study the behavior of the system in the presence of these noises, a parameter variation is added along each of these states.

Though there are fluctuations initially, it can be observed from the plots that later they are stabilized.









VI. DISCUSSION

The two link robotic arm was considered to be a mathematical model of the double pendulum. We found that as per the assumptions made and the inputs and outputs chosen, the system was controllable and observable.

The Controller/Estimator that has been designed was able to give us the desired outputs by stabilizing the system for non-linear systems as well even with the presence of external noise or disturbance. It has been observed that the system has sharp fluctuations during the stabilization process which may not be desirable in the real world application.

The motivation of the project started with the idea of designing a 6 degree of freedom robotic arm. This would involve having more states leading to higher complexity and larger estimation errors.

All the calculations were done using Matlab and Python.

REFERENCES

[1] L. MARTIN, "Control of Planar Pendulum Systems," February 2013. https://mediatum.ub.tum.de/doc/1116484/1116484.pdf

[2] A. JOSH, "Double Pendulum," May 2008. http://sophia.dtp.fmph.uniba.sk/~kovacik/doublePendulum.pdf

[3] M. DJORDJE, "Learning Motor Control for Simulated Robot Arms," University of Edinburgh, 2006. https://www.inf.ed.ac.uk/publications/thesis/online/IM060383.pdf

[4] K. J. BENJAMIN, "Double Pendulum: Construction And Exploration," Cal Poly State University, July 2012. http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1059&context=physsp