Control methods for Inverted Rotary Pendulum

Ankith Anil Das
470485327

I. Introduction

Inverted pendulum control is one of the fundamental but interesting problems in the field of control theory. This system consists of a rotary arm controlled by an input torque, τ , and an unactuated pendulum connected to the arm at a pivot joint [1]. Different control algorithms are applied for solving swing-up control and stabilization problems. The equations of motion are non-linear. Therefore, they can be linearized at an operating point to carry out proper solutions[6].

This model has been significantly used in applications such as position control, aerospace vehicle controls and robotics [2]. These applications can be found in transport machines that need to balance objects, in systems that support walking for patients, in robots that are used in domestic and industrial use and in object transport using drones [7]. This paper uses stabilization controllers to design a closed-loop pole placement.

This paper proposes design of controller for inverted pendulum system. Stabilization controllers were designed based on pole placement technique. Pole placement technique is used to get desired characteristics for the closed loop system. Pole placement is a well-known design technique that provides practical feedback gain to enable the closed-loop stable and high performance design of systems[4]. This technique also produces robustness property even in the face of certain gain[5].

The first step was to determine the equations of motion for the inverted pendulum, using the Euler-Lagrange equations. It was then continued with finding a linearized model that approximated the original nonlinear system's behavior around the equilibrium located on the upright vertical position. Using the linearized model, we designed a state feedback controller and observer by desired pole placement. Then we proceeded to make a swing up controller to bring the pendulum to a vertically upright position. Finally, we looked at a way to design a non-linear observer which can observe the system at all state values.

II. System Description

The Quanser SRV02 is a rotary servo plant attached to a pendulum arm, a physical model of such a system is presented in Fig. 1. The system introduced in this report consists of two components, a servo module and a rotary module. The servo module consists of a DC servomotor mated with a gearbox of ratio 70:1 and has an input voltage of ±6 V and is bolted directly to an aluminium frame for support. The independent output shaft driven by the motor rotates inside of a high-precision ball bearing block to minimize frictional losses. The output position and angle are measured through built in sensors on the output shaft. The second component of the system, the rotary module consists of two links, the first of which is a horizontal link called the rotating arm, and the second being a vertical link called the pendulum. The arm is rotated in the horizontal plane while the pendulum is allowed to move along the radial direction of the motor. This system was chosen as it had a detailed description of parameter values, making the modelling and simulation process more streamlined. Detailed description of the plant can be found at .... The system parameters are shown in Table. I.

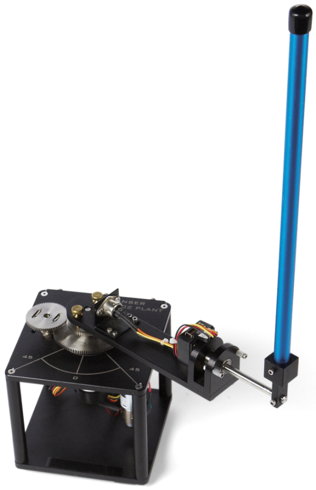

FIG. 1. Quanser SRV02 rotary servo plant attached with a pendulum arm

TABLE I. Parameters of Rotary Pendulum **(cite this)**

Parameter	Description	Value	Unit
m_r	Mass of rotary arm	0.257	kg
m_p	Mass of Pendulum	0.127	kg
L_r	Length of Rotary Arm	0.085	m
L_p	Length of Pendulum	0.129	m
J_r	Moment of Inertia rotary arm	0.0020	kgm^2
J_p	Moment of Inertia Pendulum	0.0012	kgm^2
K_t	Motor current-torque constant	0.042	Nm/A
R_m	Motor armature resistance	8.40	Ω
B_r	Viscous Damping arm	0.0015	Nms/rad
B_p	Viscous Damping Pendulum	0.0005	Nms/rad
η_m	Motor efficiency	0.69	
η_g	Gearbox efficiency	0.90	

III. System Modelling

The schematic representation of the system is shown in Fig.2. Dynamics of this system are best modelled using Lagrangian formulation of mechanics. The constrains of the system are holonomic, i.e, constrains are of the form $f(\mathbf{x}, t) = 0$ which are independent of velocity, therefore by using appropriate generalized coordinates which satisfy the constrains, lagrangian formulation can be used to model this system. Although our system would not satisfy D'Alembert's principle of virtual work, i.e, the virtual displacements are orthogonal to constrain forces, due to the presence of non-conservative forces, these non-conservative forces can be augmented to the system without affecting the Lagrangian.

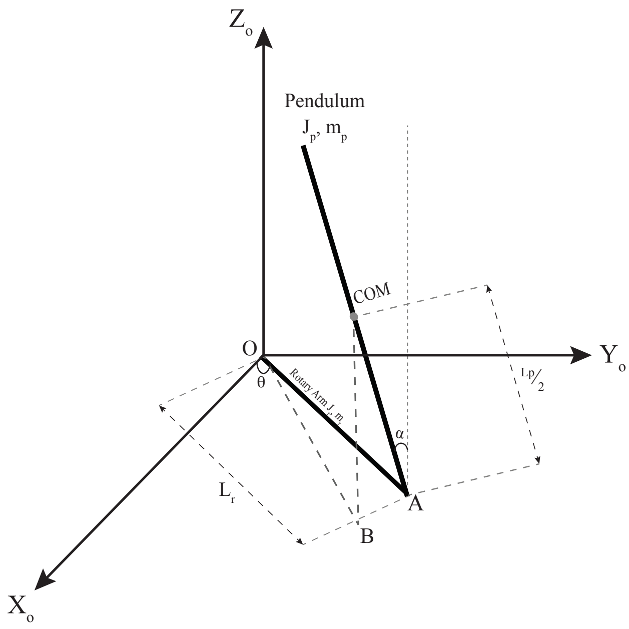

FIG. 2. Schematic representation of the plant

The generalized coordinates for the system are the angular displacement of the rotating arm θ and the pendulum angle α. The Lagrangian for a system is defined by (without non-conservative forces)

$$L = T - V \tag{1}$$

where T is the kinetic energy and V is the potential energy of the system. **Note** that the dot convention for time derivative will be used throughout this document, i.e., $\dot{\theta} = d\theta/dt$. Also t will be dropped from θ and α. The kinetic energy of rotary arm is defined by the rotational kinetic energy

$$T_{arm} = \frac{1}{2} J_r \dot{\theta}^2 \tag{2}$$

where J_r the moment of inertia of rotary arm at O. The kinetic energy of the pendulum arm is the sum rotational kinetic energy and translational kinetic energy

$$T_{pend} = \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} m_p |\dot{\mathbf{r}}_\mathbf{p}|^2 \tag{3}$$

where J_p is the moment of inertia of pendulum arm at center of mass (COM), and $\mathbf{r_p}$ is the position vector of COM of pendulum arm. Using cylindrical coordinates, the COM of pendulum arm is

$$\mathbf{r_p} = \begin{bmatrix} \sqrt{L_r^2 + \frac{L_p^2}{4} s_\alpha^2} \\ \theta - \tan^{-1}\left(\frac{L_p s_\alpha}{2L_r}\right) \\ \frac{L_p}{2} c_\alpha \end{bmatrix}_{(r,\psi,z)} \tag{4}$$

where $c_\psi = \cos\psi$ and $s_\psi = \sin\psi$. The coordinate values of radial and angular position are calculated by looking at $\triangle ABO$. The velocity is given by

$$|\dot{\mathbf{r}}_\mathbf{p}|^2 = \dot{r}^2 + r^2\dot{\psi}^2 + \dot{z}^2. \tag{5}$$

Substituting and making some cancellations give

$$|\dot{\mathbf{r}}_\mathbf{p}|^2 = \frac{1}{2} L_p^2 \dot{\alpha}^2 - L_p L_r c_\alpha \dot{\alpha}\dot{\theta} + \left(L_r^2 + \frac{1}{4} L_p^2 s_\alpha^2\right)\dot{\theta}^2. \tag{6}$$

The above equation incorporates the constrains of our system. The energy of conservative forces can be written as

$$T = \frac{1}{2} J_r \dot{\theta}^2 + \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} m_p |\dot{\mathbf{r}}_\mathbf{p}|^2, \tag{7}$$

$$V = \frac{1}{2} L_p c_\alpha. \tag{8}$$

Euler-Lagrange equations for the system are of the form

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = Q_1 \tag{9}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}}\right) - \frac{\partial L}{\partial \alpha} = Q_2 \tag{10}$$

where the generalized Q_i forces describe the non-conservative forces applied to the system with respect to the generalized coordinates. For our system, the generalized force acting on the rotary arm is

$$Q_1 = \tau - B_r \dot{\theta} \tag{11}$$

and acting on the pendulum is

$$Q_2 = -B_p \dot{\alpha} \tag{12}$$

where τ is the torque acting on the rotary arm, and B_r, B_p are viscous damping force. The control variable is the input servo motor voltage V_m $(u(t))$, and the torque generated by the servo motor is described by the equation

$$\tau = \frac{\eta K_g k_t (V_m - K_g k_m \dot{\theta})}{R_m} \tag{13}$$

Using $q = [\theta, \alpha]^T$ the terms of Lagrangian equations can be written as

$$\frac{\partial L}{\partial \dot{q}} = \begin{bmatrix} \left(J_r + m_p L_r^2 + \frac{1}{4} m_p L_p^2 s_\alpha^2\right) \dot{\theta} - \frac{1}{2} m_p L_r L_p c_\alpha \dot{\alpha} \\ \left(J_p + \frac{1}{4} m_p L_p^2\right) \dot{\alpha} - \frac{1}{2} m_p L_r L_p c_\alpha \dot{\theta} \end{bmatrix}, \tag{14}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) = \begin{bmatrix} \left(J_r + m_p L_r^2 + \frac{1}{4} m_p L_p^2 s_\alpha^2\right) \ddot{\theta} + \frac{1}{2} m_p L_p^2 c_\alpha s_\alpha \dot{\alpha}\dot{\theta} - \frac{1}{2} m_p L_r L_p c_\alpha \ddot{\alpha} + \frac{1}{2} m_p L_r L_p s_\alpha \dot{\alpha}^2 \\ \left(J_p + \frac{1}{4} m_p L_p^2\right) \ddot{\alpha} + \frac{1}{2} m_p L_r L_p s_\alpha \dot{\alpha}\dot{\theta} - \frac{1}{2} m_p L_r L_p c_\alpha \ddot{\theta} \end{bmatrix}, \tag{15}$$

$$\frac{\partial L}{\partial q} = \begin{bmatrix} 0 \\ \frac{1}{2} m_p L_p g s_\alpha + \frac{1}{2} m_p L_r L_p s_\alpha \dot{\alpha}\dot{\theta} + \frac{1}{4} m_p L_p^2 c_\alpha s_\alpha \dot{\theta}^2 \end{bmatrix}. \tag{16}$$

Finally, equations (13)-(16) can be rearranged into manipulator equation form

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu, \tag{17}$$

where,

$$H(q) = \begin{bmatrix} J_r + \frac{1}{4} m_p L_p^2 s_\alpha^2 + m_p L_r^2 & -\frac{1}{2} m_p L_p L_r c_\alpha \\ -\frac{1}{2} m_p L_p L_r c_\alpha & J_p + \frac{1}{4} m_p L_p^2 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} \frac{1}{4} m_p L_p^2 s_{2\alpha} \dot{\alpha} + \frac{\eta K_g^2 k_t k_m}{R_m} + B_r & \frac{1}{2} m_p L_r L_p s_\alpha \\ -\frac{1}{8} m_p L_p^2 s_{2\alpha} \dot{\theta} & B_p \end{bmatrix}$$

$$G(q) = \begin{bmatrix} 0 \\ -\frac{1}{2} m_p L_p g s_\alpha \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{\eta K_g K_t}{R_m} \\ 0 \end{bmatrix}$$

Note, that the choice of $C(q,\dot{q})$ is not unique. The manipulator Equation (17) can be rearranged to solve for $\ddot{q}$

$$\ddot{q} = \begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = H(q)^{-1}\left(Bu - C(q,\dot{q})\dot{q} - G(q)\right) \tag{18}$$

Using the above equation, the full non-linear system can be written out as

$$\dot{\mathbf{x}} = f(\mathbf{x}, u) \tag{19}$$

where $\mathbf{x} = [q, \dot{q}]^T$, and

$$f(\mathbf{x}, u) = \begin{bmatrix} \dot{q} \\ H(q)^{-1}\left(Bu - C(q,\dot{q})\dot{q} - G(q)\right) \end{bmatrix}$$

The full-nonlinear plant is simulated in MATLAB by integrating the above equation. The plant schematic is shown in Fig. ref.

IV. System Linearization

The linearized dynamics of any system $\dot{x} = f(x,u)$ around a equilibrium point (fixed point) (x^*, u^*) can be determined by Taylor expansion

$$\dot{x} \approx f(x^*, u^*) + \left.\frac{\partial f}{\partial x}\right|_{x^*,u^*} (x - x^*) + \left.\frac{\partial f}{\partial u}\right|_{x^*,u^*} (u - u^*)$$

therefore at fixed point

$$\dot{x} \approx \left.\frac{\partial f}{\partial x}\right|_{x^*,u^*} (x - x^*) + \left.\frac{\partial f}{\partial u}\right|_{x^*,u^*} (u - u^*).$$

Although in this example we used scalar functions f, x, u the same series expansion can be used to vector valued functions where the partial derivatives are replaced with their vector equivalent Jacobian matrices ($\nabla_{\mathbf{x}}\mathbf{f}$, $\nabla_{\mathbf{u}}\mathbf{f}$).

Even through this method of linearization can be performed for system represented in Eq.19, we won't be proceeding with this as the calculations are quite cumbersome. The tidiest way to make an approximation near an equilibrium point is to introduce new variables which are small. Thus we can write $\theta = \theta_{eq} + \theta_\epsilon$, $\alpha = \alpha_{eq} + \alpha_\epsilon$ and $u = u_{eq}$ where θ_ϵ and α_ϵ are considered to be small. The equilibrium point of the system are calculated by setting all time derivatives of Eq.17 to 0, i.e., $\dot{q} \to 0, \ddot{q} \to 0$. This gives

$$G(q) = Bu_{eq} \Rightarrow u_{eq} = 0, \alpha = 0, \pi$$

The equilibrium points of the system are

$$(\theta_{eq}, 0) \qquad\qquad (\theta_{eq}, \pi)$$

where $\theta_{eq} \in \mathbb{R}$. As expected, this system has a moving equilibrium which is independent of θ. We are interested in the dynamics of the inverted pendulum state $\alpha_{eq} = 0$. Substituting these new variables into the original ODE and keeping only linear terms gives

$$H = \begin{bmatrix} J_r + m_p L_r^2 & -\frac{1}{2} m_p L_p L_r \\ -\frac{1}{2} m_p L_p L_r & J_p + \frac{1}{4} m_p L_p^2 \end{bmatrix} \tag{20}$$

$$C = \begin{bmatrix} \frac{\eta K_g^2 k_t k_m}{R_m} + B_r & 0 \\ 0 & B_p \end{bmatrix} \tag{21}$$

$$G = \begin{bmatrix} 0 \\ -\frac{1}{2} m_p L_p g \alpha_\epsilon \end{bmatrix} \tag{22}$$

Here the approximation $\sin\alpha_\epsilon \approx \alpha_\epsilon$ and $\cos\alpha_\epsilon \approx 1$ was used to simplify the equations. Solving for acceleration terms, we get

$$\begin{bmatrix} \ddot{\theta}_\epsilon \\ \ddot{\alpha}_\epsilon \end{bmatrix} = H^{-1} \begin{bmatrix} \frac{\eta K_g k_t (u - K_g k_m \dot{\theta}_\epsilon)}{R_m} - B_r \dot{\theta}_\epsilon \\ \frac{1}{2} m_p L_p g \alpha_\epsilon - B_p \dot{\alpha}_\epsilon \end{bmatrix}. \tag{23}$$

The above equations are linear (H is a constant matrix) and therefore can be represented in state space form.

Moving back to original variables (θ, α) and expanding the above matrix gives

$$\ddot{\theta} = \frac{1}{\Delta}\left(-\left(J_p + \frac{1}{4}m_pL_p^2\right)\left(\frac{\eta K_g^2 k_t k_m}{R_m} + B_r\right)\dot{\theta} - \frac{1}{2}m_pL_pL_rB_p\dot{\alpha} + \frac{1}{4}m_p^2L_p^2L_rg\alpha + \left(J_p + \frac{1}{4}m_pL_p^2\right)\frac{\eta K_g k_t u}{R_m}\right) \tag{24}$$

$$\ddot{\alpha} = \frac{1}{\Delta}\left(-\frac{1}{2}m_pL_pL_r\left(\frac{\eta K_g^2 k_t k_m}{R_m} + B_r\right)\dot{\theta} - \left(J_r + m_pL_r^2\right)B_p\dot{\alpha} + \frac{1}{2}m_pL_pg\left(J_r + m_pL_r^2\right)\alpha + \frac{1}{2}m_pL_pL_r\frac{\eta K_g k_t u}{R_m}\right) \tag{25}$$

$$A = \frac{1}{\Delta}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{4}m_p^2L_p^2L_rg & -\left(J_p + \frac{1}{4}m_pL_p^2\right)\left(\frac{\eta K_g^2 k_t k_m}{R_m} + B_r\right) & -\frac{1}{2}m_pL_pL_rB_p \\ 0 & \frac{1}{2}m_pL_pg\left(J_r + m_pL_r^2\right) & -\frac{1}{2}m_pL_pL_r\left(\frac{\eta K_g^2 k_t k_m}{R_m} + B_r\right) & -\left(J_r + m_pL_r^2\right)B_p \end{bmatrix} \tag{26}$$

$$B = \frac{\eta K_g k_t}{\Delta R_m}\begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_pL_p^2 \\ \frac{1}{2}m_pL_pL_r \end{bmatrix} \tag{27}$$

where $\Delta = \det(H)$. After collecting the terms and arranging, we get the above state space matrices. Some of the computations above were preformed using *Mathematica* as it was too cumbersome and error prone to be performed by hand. After substituting parameter values from Table. I, the state space matrices are

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 12.27 & -8.67 & -0.0764 \\ 0 & 51.44 & -3.49 & -0.32 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 15.07 \\ 6.0715 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

By computing the reachability matrix, we see that the system is controllable. Also, we see that the system is unstable as A has positive eigenvalues. We also see that the system is observable as the observability matrix is full rank. To see how well the linearized system performs, the linear and non linear system with step impulse is plotted in Fig.3

V. Controller Design and Simulations

In this section, we will be discussing and designing these components

- A full state feedback controller for stabilizing inverted position
- A state observer to estimate state and attenuate signal noise
- A Swing up controller

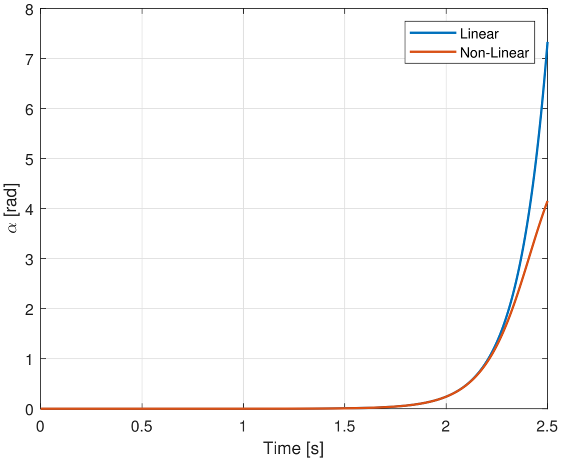

FIG. 3. Comparison of Linear and Non-Linear system

- A Non-Linear observer to observe the full state of the system

A. Full state feedback controller

The requirements of full state feedback controller are:

1. Smoothly and precisely achieve any rotary angle θ
2. Maintain stability in the presence of noisy signals

The primary considerations for this system is to determine the expected system characteristics like,overshoot, rise time, and settling time. As this system is quite unstable and can enter non-linear region quite easily, the controller must have low overshoot with a sufficiently large rise time and settling time. Rapid changes in angles or controller effort can push the system to non-linear region and would result the pendulum falling off.

The maximum overshoot for the system was determined to be 5%. Having a low overshoot will reduce the overall controller effort and provide a smooth transition from one reference point to another. The settling time is an important factor for stability. A quick stabilization time of < 2 would provide quick response as well as a fast stabilization. Finally, we also require that the pendulum angle $|\alpha| < 15°$ and the controller effort is less than the maximum voltage $|V_m| < 6\text{V}$.

With the required characteristics of the system outlined, we can calculated the desired pole locations. The percentage of overshoot is calculated by

$$M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}} \tag{28}$$

where ζ is the damping constant of the system. A 5% overshoot corresponds to

$$\zeta = \frac{1}{\sqrt{2}} \approx 0.7 \tag{29}$$

With a settling time $T_s = 2$ seconds, the natural frequency of the system can be approximated

$$T_s = \frac{4}{\zeta\omega_n} \Rightarrow \omega_n = 2.857\text{rad/s} \tag{30}$$

From this, the desired dominant poles are

$$p_{1,2} = -\sigma \pm j\omega_d \tag{31}$$

where $\sigma = \zeta\omega_n$ and $\omega_d = \omega_n\sqrt{1-\zeta^2}$. Substituting the above values,we get the following two dominant poles

$$p_{1,2} = -2 \pm 2.04j \tag{32}$$

The non dominant poles of this system are set far from the desired poles so that they have no significant impact on the system response. The non-dominant poles were set to $p_{3,4} = -30, -40$. The controller gains can be easily calculated using MATLAB 'place' functions which will calculate the required gains so $A - BK$ have the desired eigenvalues. These poles are placed by optimizing the choice of eigenvectors for a robust solution(cite this). The controller gains are

$$K = \begin{bmatrix} -13.97 & 293.83 & -8.32 & 31.38 \end{bmatrix} \tag{33}$$

Finally, the closed loop response of the non-linear plant with the controller is shown in Fig.4. In this simulation, a reference generator is used which change $\theta = [30, -60, 45, -30]°$ once it has stabilized (threshold of $3°$) at a position for more than 3 seconds. The step output from the reference generator is passed through a 1^{st} order low pass filter to smooth out the signal. The controller effort during this simulation is well below the threshold and is shown in Fig. 5. The system is also able to maintain the pendulum upright when there a signal noise of 10^{-2} order. This was tested by adding white noise to the output signal and is shown in Fig.6.

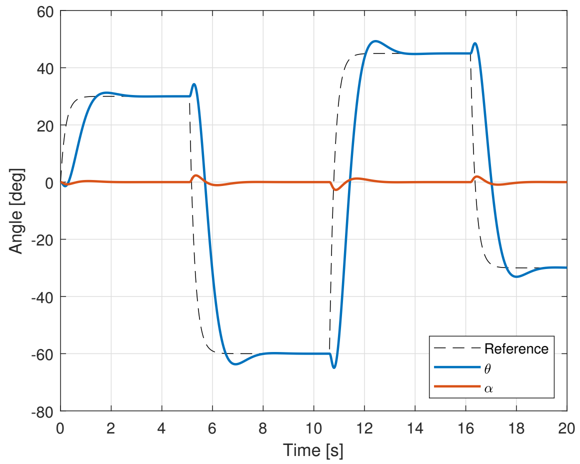

FIG. 4. Closed loop response of the Non-linear plant

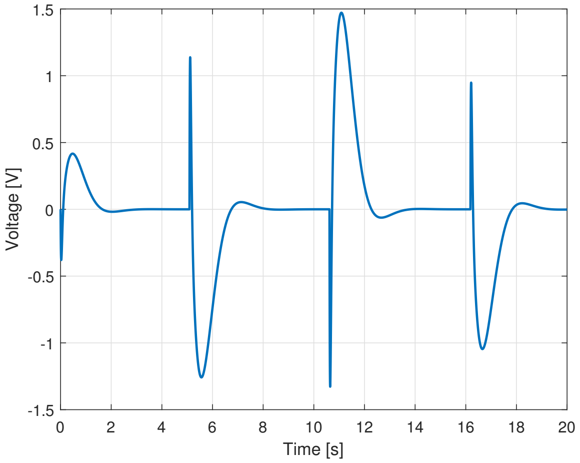

FIG. 5. Closed loop controller output

B. Designing a Linear state estimator

As our system does not have dedicated sensors to measure the angular velocity of the pendulum or the arm, we need to estimate these states with knowledge of current position and controller input. One novel but an inefficient solution would be time derivate of position to get angular velocity. Even though in an idealized situation, this would be feasible, but most sensor outputs have noise. Therefore, a linear state estimator with noise attenuation is required.

A general linear observer have a form

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) \tag{34}$$

where L is the observer gains and $\hat{x}$ is the estimated state. The estimated error $\tilde{x} = x - \hat{x}$ would have the dynamics

$$\frac{d\tilde{x}}{dt} = (A - LC)\tilde{x}. \tag{35}$$

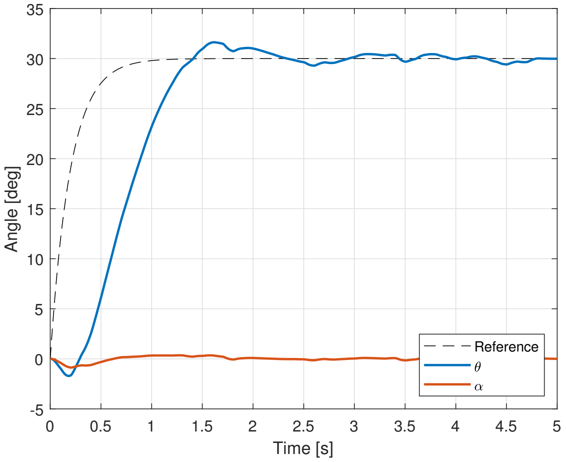

FIG. 6. Closed loop response with white noise of order 10^{-2} added to output signal

Then by choosing L such that $A - LC$ have eigenvalues with negative real parts will ensure that the error will converge to 0. The convergence rate is determined by the appropriate selection of L. Choosing large negative eigenvalues will allow the observe to converge quickly, but will also be sensitive to noise. Therefore, the gains are selected such that estimator converges about twice as fast as the dynamics of the closed loop system. This allows a good estimate of the current state at the same time being insensitive to signal noise.

With the above fact taken into considerations, the closed loop poles of the observer are selected beyond $\Re(p) = -4$. The selected poles are

$$p = \begin{bmatrix} -4 & -4.5 & -5 & -5.5 \end{bmatrix}. \tag{36}$$

Using the poles, calculated observer gains are

$$L = \begin{bmatrix} 1.37 & -0.134 \\ -2.72 & 8.64 \\ 13.26 & 12.51 \\ -0.296 & 68.95 \end{bmatrix}. \tag{37}$$

Using this observer, we can use the same controller designed in the previous section as a consequence of separation principle. This allows up to independently assign eigenvalues for a controller and observer and combine them together. The full Simulink system is shown in Fig. 7. Even with large signal noise added to the system, the observer is able to estimate the states accurately which allows controller to move the pendulum accurately. The output signal y and the estimated value is shown in Fig. 8. As we can see, the system is still being able to maintain the pendulum vertical and track quite accurately. The system does spend more time at $\theta = 45°$ because it fails to maintain a stable angle with a threshold of 3° for more than 3 seconds. Finally, the controller output is checked and it is also well below the 6 V threshold.

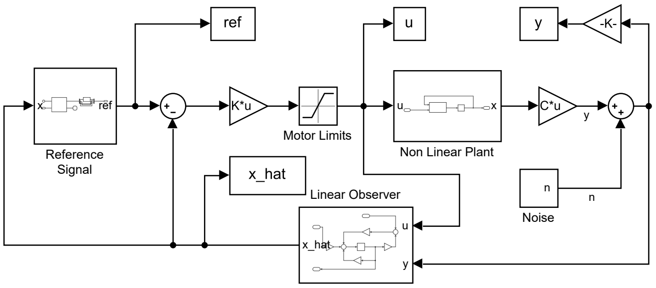

FIG. 7. Closed loop controller output

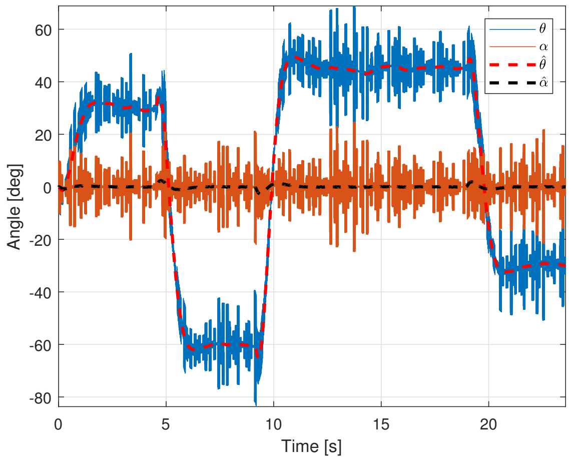

FIG. 8. Comparison between output signal y and estimated state $\hat{x}$. Here signal noise

C. Swing up control

The objective of this controller is to bring the pendulum from downward position to vertically upright position, then the full state feedback controller can stabilize the pendulum at this state. A switching logic is used to switch over from swing up control to stabilization control.

The swing up action is attained using an energy controller. Here the total energy of the system is used as a feedback quantity. It is driven such a way that the system achieves the energy equal to the upright position. By neglecting frictional losses, we can write refine the dynamics of the pendulum in terms of pivot acceleration as

$$J_p \ddot{\alpha} = \frac{1}{2} m_p L_p u \cos\alpha + \frac{1}{2} m_p L_p g \sin\alpha \tag{38}$$

where u is the pivot acceleration. The energy of the pendulum in pivot frame is

$$E = \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} m_p g L_p \cos\alpha. \tag{39}$$

Taking time derivative of the above equation and simplifying gives

$$\dot{E} = \frac{1}{2} m_p L_p u \dot{\alpha} \cos\alpha \tag{40}$$

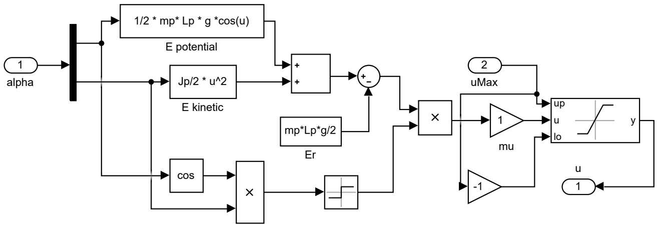

FIG. 9. Energy controller subsystem. This subsystem takes $\alpha, \dot{\alpha}$ and calculates the required pivot velocity u

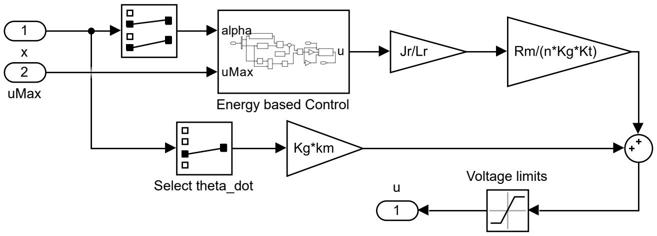

FIG. 10. Swing Up Controller. This system takes $\mathbf{x}$ and calculates the controller output voltage

From the above equation, we can see that the energy of the pendulum is simply an integrator with varying gains. A control strategy where

$$u = (E - E_r)\, \dot{\alpha} \cos \alpha \tag{41}$$

would drive the system to the desired energy E_r. The above control law is non-linear and depends on pendulum angle α and $\dot{\alpha}$. From Eq. 39, we can see that required reference energy for the pendulum to be upright is $1/2\ m_p L_p g$J.

As the system has damping forces, it is necessary to change the energy quickly. We can modify the control law to

$$u = \mathrm{sat}_{u_{\max}} \left(\mu\, (E - E_r)\, \mathrm{sign}(\dot{\alpha} \cos \alpha)\right) \tag{42}$$

where μ is a controller gain, $\mathrm{sat}_{u_{\max}}$ function saturates the control signal's maximum acceleration. This controller basically acts like a bang-bang controller for large errors and as a proportional control for small errors.

Finally, the pivot acceleration is transformed into servomotor voltage to get

$$V_m = \frac{R_m J_r u}{\eta_g K_g \eta_m L_r k_t} + K_g k_m \dot{\theta} \tag{43}$$

where $\tau = J_r/L_r$.

The energy control subsystem is shown in Fig. 9. This subsystem encapsulates Eq.42. The controller output voltage is calculated by the Swing up controller shown in 10.

To combine the swing up controller with the feedback controller, a switching logic is used which switches control over to the feedback controller when the pendulum is vertical. A function block is used that converts $\alpha \in (-\pi, \pi)$ so that the switch and controller works as expected. The switch logic can be summarized as

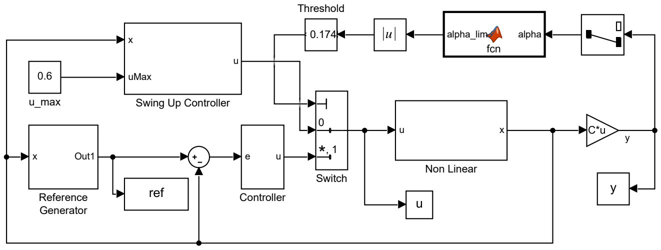

FIG. 11. Swing up system

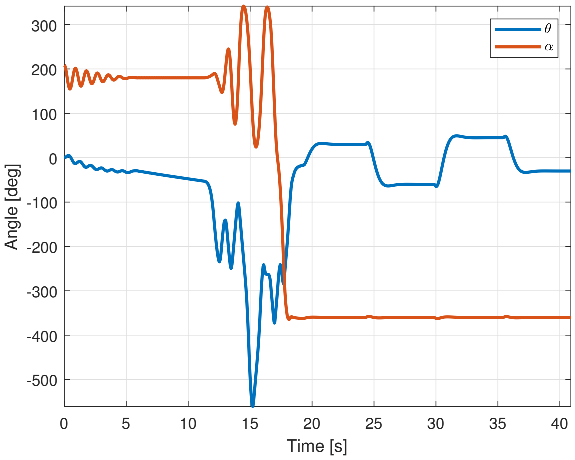

FIG. 12. Swing up action of the Pendulum

$$u = \begin{cases} K(\mathbf{r} - \mathbf{x}) & |\alpha| < 10^\circ \\ \mathrm{sat}_{u_{\max}} \left(\mu\, (E - E_r)\, \mathrm{sign}(\dot{\alpha} \cos \alpha)\right) & \text{Otherwise.} \end{cases} \tag{44}$$

The combined system is shown in Fig.11.

To see the controller in action, the pendulum was given a small oscillation in the downward position. This is necessary as $\dot{\alpha} \cos \alpha$ would be zero, resulting in zero controller output. Once there is small oscillations, the pendulum starts swinging until it becomes vertical. The system in action is shown in Fig. 12. Once the pendulum is vertical and stable, the reference generator moves the pendulum to different set points.

Although the controller makes the pendulum vertical, it doesn't perform it with minimum energy. The controller often exerts large effort which results in multiple pendulum windups. Also, the rotary arm moves large angles to swing up the pendulum. The controller effort is shown in Fig.13. As we can see, the controller effort saturates just after the swing up action. This unusual behavior is seen just after the swing up controller gives control to the feedback controller. Due to the large input, the pendulum swings over and control is regained after the second swing. This can be seen in 12 where the controller fails to maintain vertical position when $\alpha = 0$.

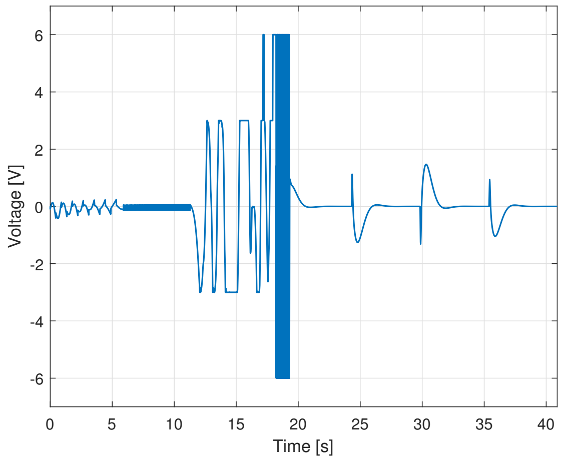

FIG. 13. Controller effort during swing up action

Changing the switch threshold angle did not fix this issue.

D. Designing a Non-Linear Observer

During the swing up action or when the pendulum is down, the linear observer designed in the previous section cannot be used to estimate the full state of the system. This is because, the states are very far from the linearization point. To observe these states, a non-linear observer is proposed. The idea behind this is to design a time varying gain $L(\hat{\mathbf{x}})$ such that

$$\frac{d\hat{x}}{dt} = f(\hat{\mathbf{x}}, u) + L(\hat{\mathbf{x}})(y - C\hat{x}) \tag{45}$$

converges to the true state. Here, $f(\mathbf{x}, u)$ is the full non-linear system described in Eq. 19.

To calculate $L(\hat{\mathbf{x}}, t)$, the system is linearized at the estimated operating point

$$f(\hat{\mathbf{x}}, u) \approx f\left(\hat{\mathbf{x}}^*, u^*\right) + \nabla_{\mathbf{x}} \mathbf{f}_{x^*,u^*}\left(\hat{\mathbf{x}} - \hat{\mathbf{x}}^*\right) + \nabla_{\mathbf{u}} \mathbf{f}_{x^*,u^*}\left(u - u^*\right) \tag{46}$$

where $\hat{\mathbf{x}}^*, u^*$ is the current estimated state and $\hat{\mathbf{x}}, u$ is the next estimated state. By using $A := \nabla_{\mathbf{x}} \mathbf{f}$ and $B := \nabla_{\mathbf{u}} \mathbf{f}$, the observer gain are calculated such that the eigenvalues of $A - LC$ have negative real parts. This allows the observer converge to the true state of the system.

The analytical expression for A and B were calculated in *Mathematica* and then imported as a matrix expression to Matlab. These expressions are too long to be expressed in this report. Using these matrices, the observer gains are calculated by using a modified 'place' function. Some modifications were needed to the original 'place' function to make it compatible for code generation. The same observer poles were that was used for linear observer was used to calculate $L(\hat{\mathbf{x}})$. The schematic of the full-state observer is shown in Fig.14

To test this observer, open loop response of the system and the observer were compared when the vertical pendulum was disturbed with a small impulse voltage. The

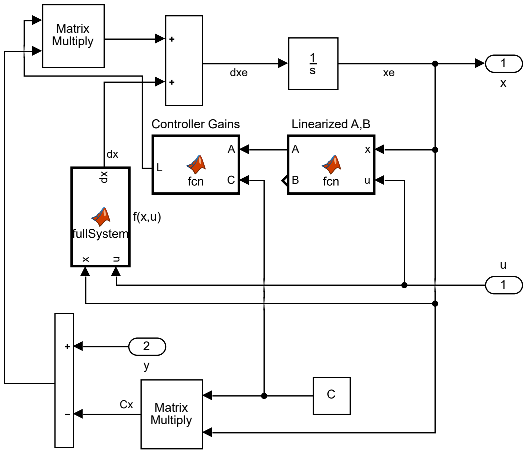

FIG. 14. Schematic representation of Non-Linear Observer.

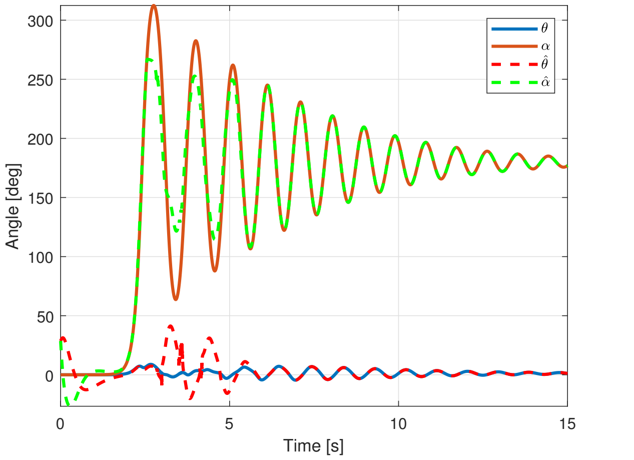

FIG. 15. Non Linear observer converging to state angular position when the pendulum falls down.

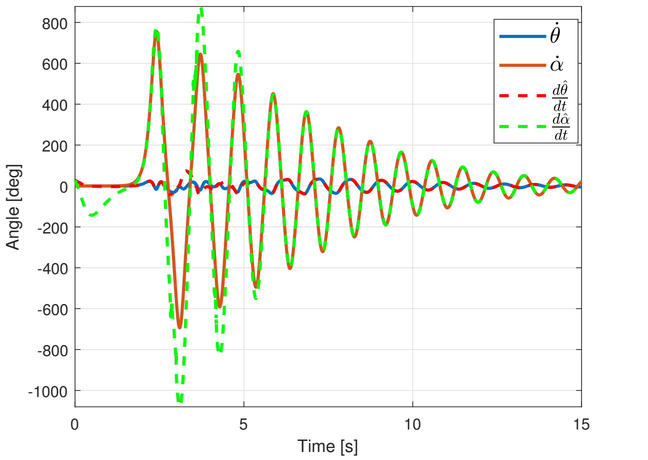

FIG. 16. Non Linear observer converging to state angular velocity when the pendulum falls down.

observer was initialed at $\mathbf{x} = [0.5, 0.5, 0.5, 0.5]^T$ to see the convergence of the estimated state and actual state. Fig.15, 16 shows how the estimator converges to the true state.

Unfortunately, adding the non-linear observer to the swing up system causes convergence error which crashes the simulation. The automatically selected solver, ode15s crashes when the swing up controller outputs a large voltage. Using a different solver like ode45 did provides an output, but the swing up action was lost. This could be because of the non-linear nature of the swing up controller. The schematic represented of this system is shown in Fig.17.

Even though the swing up action failed when ode45 was used, the estimated state and actual state were compared to see the performance of the observer. The observer was initialized at $\mathbf{x} = [0.5, 0.5, 0.5, 0.5]^T$ and the pendulum system at $\mathbf{x} = [0, \pi + 0.5, 0, 0]^T$ and the result is shown in Fig.18, 19

VI. Conclusion

In the report, we went through the steps of necessary to design controllers for inverted rotary pendulum. We began by modelling our system using lagrangian formulation of mechanics. After we developed our non-linear model, we linearized the model at an equilibrium point to get state space matrices. Using these matrices, we designed a state feedback controller and observer by desired pole placement. Then we proceeded to make a swing up controller to bring the pendulum to a vertically upright position. Finally, we looked at a way to design a non-linear observer which can observe the system at all state values. We also tested the performance of this observer in the full system. Although the swing up controller was able to bring the pendulum to vertical position, it caused some unexpected behavior which could not be explained. Adding the non-linear observer to the swing up system causes the simulation to fail due to convergence error. Using other solves like ode45 did produce a plot, but the swing up action was lost. This could be due to non-linear nature of the controller. Further investigation is needed to identify this issue.

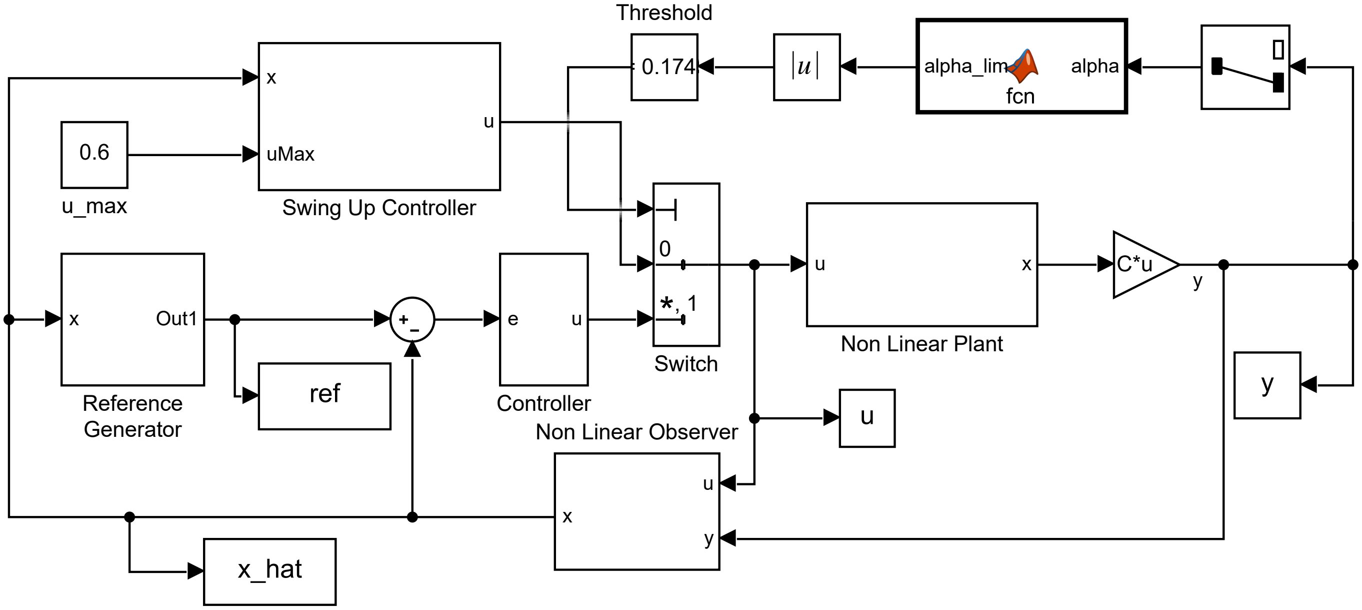

FIG. 17. Swing up system with non-linear observer

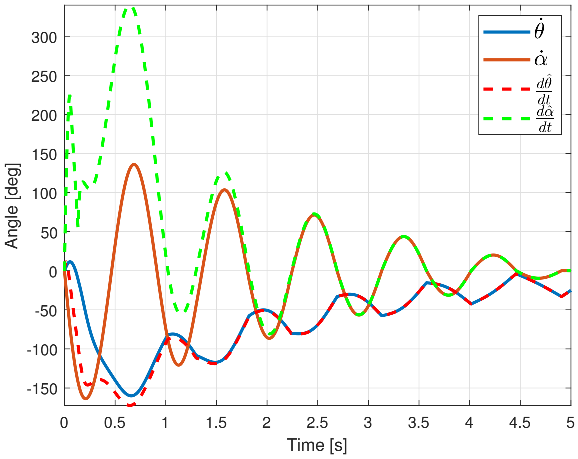

FIG. 18. Non-linear estimator converging to true state angular velocity during a failed swing up.

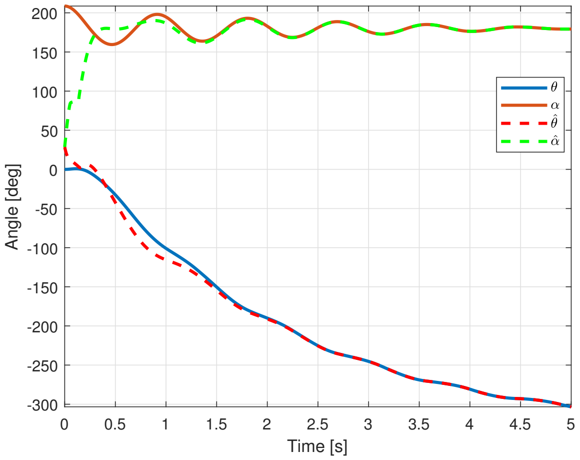

FIG. 19. Non-linear estimator converging to true state angular position during a failed swing up.