NVEO
Natural Volatiles & Essential Oils

Adaptive Dynamic Programming Based Linear Quadratic Regulator Design for Rotary Inverted Pendulum System

Gavtham Hari Kumar B[#1], Vimal E[#2]

Department of Instrumentation and Control Systems Engineering,
PSG College of Technology, Coimbatore, India.
[#1]UG Scholar, [#2]Assistant professor
#[1]gavthambabu@gmail.com
#[2]evm.ice@psgtech.ac.in

Abstract

The rotary inverted pendulum system is an inherently unstable system with highly nonlinear dynamics. It is used for design, testing, evaluating and comparing of different classical and contemporary control techniques. The goal of this project is to design an ADP based LQR controller for the rotary inverted pendulum system. Here model-based policy iteration algorithm is used to design the ADP based LQR controller. The swing-up and balance control is also implemented for the rotary inverted pendulum system using ADP based LQR controller gain. The response of the rotary inverted pendulum system with conventional LQR controller, ADP based LQR controller, swing-up and balance control is illustrated using MATLAB–SIMULINK platform. The result obtained after comparing the ADP based LQR controller response with conventional LQR controller, the rotary inverted pendulum system is stabilized faster with ADP based LQR controller and the swing-up and balance control response of the rotary inverted pendulum system has also improved due to ADP based LQR controller gain.

Keywords*—ADP, LQR controller, Swing up and balance control.*

I. INTRODUCTION

The inverted pendulum is an inherently unstable system with highly nonlinear dynamics. This is a system which belongs to the class of under-actuated mechanical systems having fewer control inputs than the degree of freedom. This renders the control task more challenging, making the inverted pendulum system a classical benchmark for the design, testing, evaluating and comparing of different classical and contemporary control techniques. Being an inherently unstable system, the inverted pendulum is among the most difficult systems, and is one of the most important classical problems.

The numerous practical applications of the rotary inverted pendulum system make its study pertinent. In robotics, balancing systems are developed using inverted pendulums. These find application 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. Therefore, controlling this system is essential and throughout the years many classical control solutions are proposed. However, for more efficient control this project proposes an ADP based LQR controller for controlling the rotary inverted pendulum system.

H. Wang, H. Dong, L. He, Y. Shi and Y. Zhang, "Design and Simulation of LQR Controller with the Linear Inverted Pendulum," International Conference on Electrical and Control Engineering, vol. 2, pp. 699-702, 2010 - This paper focused on modelling and performance analysis of linear inverted pendulum and design and simulation of LQR controller. Main to introduce how to build the mathematic model and the analysis of its system performance, then design a LQR controller in order to get the much better control. Simulation is done to show the efficiency and feasibility of proposed approach [1].

F. A. Yaghmaie and S. Gunnarsson, "A New Result on Robust Adaptive Dynamic Programming for Uncertain Partially Linear Systems," IEEE 58th Conference on Decision and Control (CDC), vol. 71, pp. 7480-7485, 2019 - This paper, presents a new result on robust adaptive dynamic programming for the Linear Quadratic Regulation (LQR) problem, where the linear system is subject to unmatched uncertainty. They assume that the states of the linear system are fully measurable and the matched uncertainty models unmeasurable states with an unspecified dimension. They used the small-gain theorem to give a sufficient condition such that the generated policies in each iteration of on-policy and off-policy routines guarantee robust stability of the overall uncertain system. The sufficient condition can be used to design the weighting matrices in the LQR problem and simulation example are given to demonstrate the result [2].

Y. Liu, Y. Luo and H. Zhang, "Adaptive dynamic programming for discrete-time LQR optimal tracking control problems with unknown dynamics," IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), vol. 9, pp.1-6, 2014 – In this paper, an optimal tracking control approach based on adaptive dynamic programming (ADP) algorithm is proposed to solve the linear quadratic regulation (LQR) problems for unknown discrete-time systems in an online fashion. First, we convert the optimal tracking problem into designing infinite-horizon optimal regulator for the tracking error dynamics based on the system transformation. Then we expand the error state equation by the history data of control and state. The iterative ADP algorithm of PI and VI are introduced to solve the value function of the controlled system. It is shown that the proposed ADP algorithm solves the LQR without requiring any knowledge of the system dynamics. The simulation results show the convergence and effectiveness of the proposed control scheme [3].

S. A. A. Rizvi and Z. Lin, "Reinforcement Learning-Based Linear Quadratic Regulation of Continuous-Time Systems Using Dynamic Output Feedback," in IEEE Transactions on Cybernetics, vol. 50, pp. 4670-4679, 2020 - In this paper, we propose a model-free solution to the linear quadratic regulation (LQR) problem of continuous-time systems based on reinforcement learning using dynamic output feedback. The design objective is to learn the optimal control parameters by using only the measurable input-output data, without requiring model information. A state parametrization scheme is presented which reconstructs the system state based on the filtered input and output signals. Based on this parametrization, two new output feedback adaptive dynamic programming Bellman equations are derived for the LQR problem based on PI and VI. Unlike the existing output feedback methods for continuous-time systems, the need to apply discrete approximation is obviated. In contrast with the static output feedback controllers, the proposed method can also handle systems that are state feedback stabilizable but not static output feedback stabilizable. An advantage of this scheme is that it stands immune to the exploration bias issue. Moreover, it does not require a discounted cost function and, thus, ensures the closed-loop stability and the optimality of the solution. Compared with earlier output feedback results, the proposed VI method does not require an initially stabilizing policy. We show that the estimates of the control parameters converge to those obtained by solving the LQR algebraic riccati equation. A comprehensive simulation study is carried out to verify the proposed algorithms [4].

II. System Description

A. QUANSER QUBE-SERVO 2

The QUANSER QUBE-Servo 2, pictured in Figure 1, is a compact rotary servo system that can be used to perform a variety of classic servo control and inverted pendulum-based experiments. The QUBE-Servo 2 comes in three versions: the USB Interface, Direct I/O Interface, and NI myRIO Interface. The QUBE-Servo 2 USB Interface has its own built-in power amplifier and data acquisition device. The QUBE-Servo 2 Direct I/O Interface also has an integrated amplifier but allows an external DAQ device to interface to its I/O. The QUBE-Servo 2 myRIO Interface also has a built-in amplifier, and allows a direct connection to the NI MXP connector. For all versions, the system is driven using a direct-drive 18V brushed DC motor housed in a solid aluminium frame. Two add-on modules are supplied with the system: an inertial disc and a rotary pendulum. The modules can be easily attached or interchanged using magnets mounted on the QUBE-Servo 2 module connector. Single-ended rotary encoders are used to measure the angular position of the DC motor and pendulum.

Main QUBE-Servo 2 features:

- Compact and complete rotary servo system
- 18V direct-drive brushed DC motor
- Encoders mounted on DC motor and pendulum
- Built-in PWM amplifier
- Built-in USB DAQ device (only for QUBE-Servo 2 USB Interface)
- Inertial disc module
- Rotary pendulum module [5].

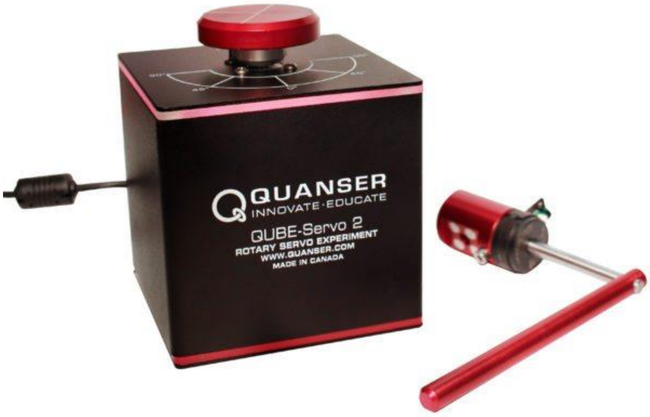


Fig. 1 QUANSER QUBE-Servo 2 Rotary Inverted Pendulum System

B. Hardware Components

The main QUBE-Servo 2 components are listed in table I. The components on the QUBE-Servo 2 USB Interface are labelled in figure 2(a), the components on the QUBE-Servo 2 Direct I/O Interface are shown in figure 2(b), and the components on the QUBE-Servo 2 myRIO Interface are in figure 2(c). The interaction between QUANSER QUBE-Servo 2 components is also shown in figure 3.

TABLE I

QUBE-SERVO 2 COMPONENTS

ID	COMPONENTS	ID	COMPONENTS
1	Chassis	11	Rotary arm hub
2	Module connector	12	Rotary pendulum magnets
3	Module connector magnets	13	Pendulum encoder
4	Status LED strip	14	DC motor
5	Module encoder connector	15	Motor encoder
6	Power connector	16	QUBE-Servo 2 DAQ/amplifier board
7	System power LED	17	SPI Data Connector
8	Inertia disc	18	USB connector
9	Pendulum link	19	Interface power LED
10	Rotary arm rod	20	Internal data bus

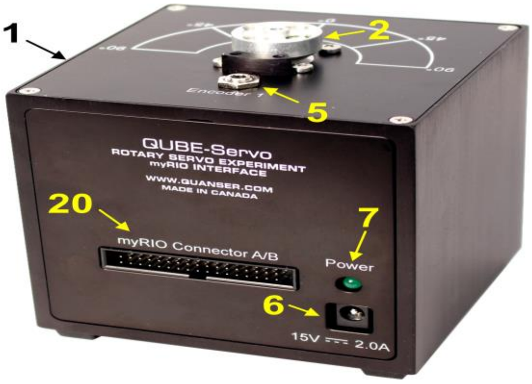
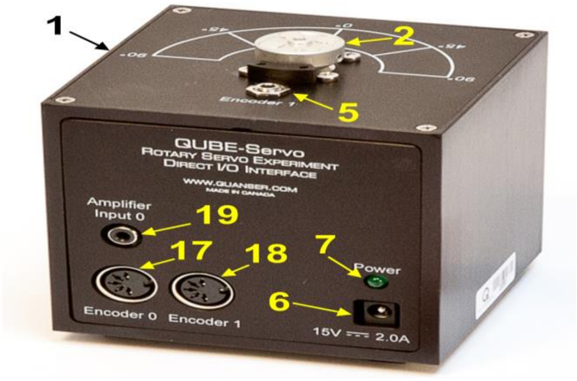


Fig. 2(a) QUBE-Servo 2 USB Interface



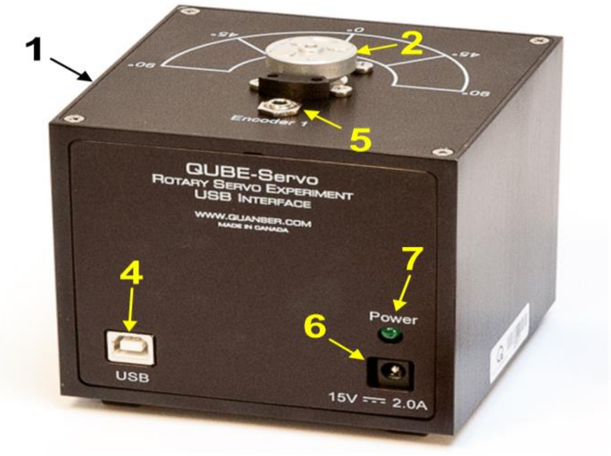


Fig. 2(b) QUBE-Servo 2 Direct I/O Interface

Fig. 2(c) QUBE-Servo 2 myRIO Interface

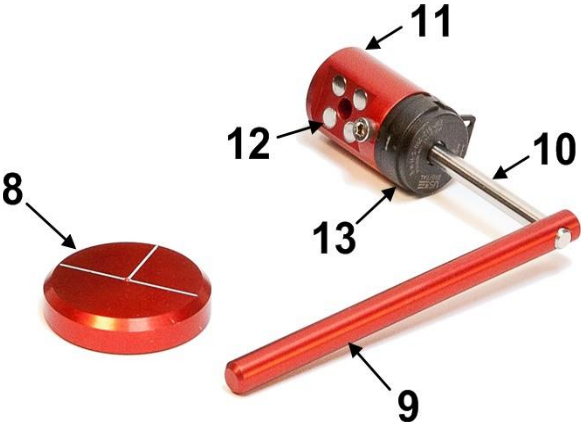


Fig. 2(d) QUBE-Servo 2 Modules

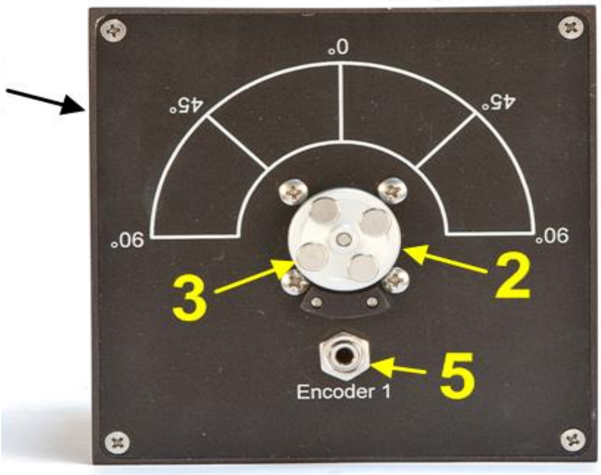


Fig. 2(e) QUBE-Servo 2 Top View

1) DC Motor

The QUBE-Servo 2 includes a direct-drive 18V brushed DC motor. The motor specifications are given in table II.

2) Encoder

The encoder used to measure the angular position of the DC motor and pendulum on the QUBE-Servo 2 is a single ended optical shaft encoder. It outputs 2048 counts per revolution in quadrature mode (512 lines per revolution). A digital tachometer is also available for angular speed in counts/sec on channel 14000.

The encoder used to measure the angular position of the DC motor and pendulum on the QUBE is the US Digital E8P-512-118 single-ended optical shaft encoder. The complete specification sheet of the E8P optical shaft encoder is given in E8P Data Sheet.

3) Data acquisition (DAQ) device

The QUBE-Servo 2 includes an integrated data acquisition device with two 24-bit encoder channels with quadrature decoding and one PWM analog output channel. The DAQ also incorporates a 12-bit ADC which provides current sense feedback for the motor. The current feedback is used to detect motor stalls and will disable the amplifier if a prolonged stall is detected.

4) Power Amplifier

The QUBE-Servo 2 circuit board includes a PWM voltage-controlled power amplifier capable to providing 2A peak current and 0.5A continuous current (based on the thermal current rating of the motor). The output voltage range to the load is between ±10 V.

Amplifier Input Connector

The amplifier input RCA connector on the QUBE-Servo Direct I/O Interface is shown in figure 2(b). It is single ended and has a range of 10V. As shown in figure 3, it is connected to the amplifier command which then drives the motor.

5) Encoder Connector

The Encoder 0 and Encoder 1 5-pin DIN connectors pictured on the QUBE-Servo Direct I/O interface in figure 2(b) output the measurements from the motor encoder and the add-on module (e.g., pendulum) encoder, respectively.

6) MXP Connector

The myRIO Connector A/B connector pictured on the QUBE-Servo myRIO Interface in figure 2(c) is used to connect the amplifier command line, and encoder readings from the QUBE-Servo components to either of the two NI myRIO MXP connectors [5].

C. System Parameters

TABLE II

QUANSER QUBE-SERVO 2 ROTARY INVERTED PENDULUM SYSTEM PARAMETERS

DC Motor		
V_{nom}	Nominal input voltage	18.0 V
τ_{nom}	Nominal torque	22.0 mN-m
ω_{nom}	Nominal speed	3050 RPM
I_{nom}	Nominal current	0.540 A
R_m	Terminal resistance	8.4 Ω
k_t	Torque constant	0.042 N-m/A
k_m	Motor back-emf constant	0.042 V/(rad/s)
J_m	Rotor Inertia	4.0×10^{-6} kg-m^2
L_m	Rotor inductance	1.16 Mh
m_h	Module attachment hub mass	0.0106 kg
r_h	Module attachment hub radius	0.0111 m
J_h	Module attachment moment of Inertia	0.6×10^{-6} kg- m^2
Inertia Disc Module		
m_d	Disc mass	0.053 kg
r_d	Disc radius	0.0248 m
Rotary Pendulum Module		
m_r	Rotary arm mass	0.095 kg
L_r	Rotary arm length (pivot to end of metal rod)	0.085 m
m_p	Pendulum link mass	0.024 kg
L_p	Pendulum link length	0.129 m

D. State Space Model of Rotary Inverted Pendulum

1) DC Motor Modelling

This section summarizes how to find the equations of motion of the DC motor. The motor electrical equation is

$$v_m(t) - R_m i_m(t) - k_m \dot{\theta}_m(t) = 0 \qquad (1)$$

where $v_m(t)$ is the motor input voltage (the control input), R_m is the motor electrical resistance, $i_m(t)$ is the current, k_m is the back-emf constant, and $v_m(t)$is the angular position of the motor shaft (i.e., the inertia disc).The motor shaft equation is expressed as

$$J_{eq}\ddot{\theta}(t) = \tau_m(t) \qquad (2)$$

where J_{eq} is the total or equivalent moment of inertia acting on the motor shaft and τ_m is the applied torque from the DC motor. Based on the current applied, the torque is

$$\tau_m(t) = k_t\, i_m(t) \qquad (3)$$

where k_t is the motor current torque constant [5].

2) Rotary Pendulum Model

The rotary pendulum model is shown in figure 2.4. The rotary arm pivot is attached to the QUBE Servo 2 system and is actuated. The arm has a length of r, a moment of inertia of J_r, and its angle θ increases positively when it rotates counter clockwise. The servo (and thus the arm) should turn in the CCW direction when the control voltage is positive, $v_m > 0$.

The pendulum link is connected to the end of the rotary arm. It has a total length of L_P and it center of mass is at $I = L_P/2$. The moment of inertia about its center of mass is J_p. The rotary pendulum angle α is zero when it is hanging downward and increases positively when rotated CCW [5].

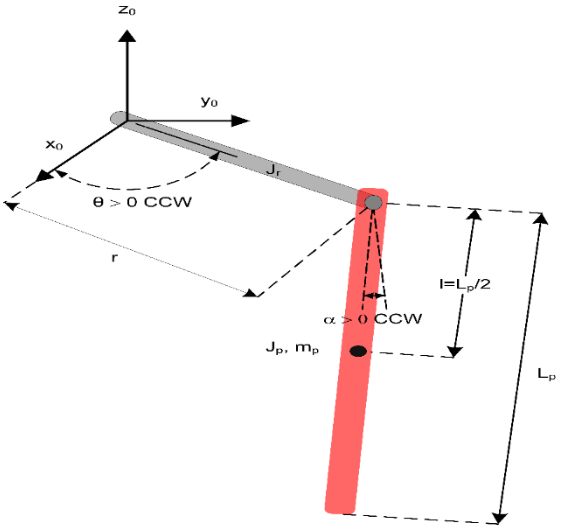


Fig. 4 Rotary Pendulum Model

The equations of motion for the pendulum system were developed using the Euler LaGrange method. This systematic method is often used to model complicated systems such as robot manipulators with multiple joints. The total kinetic and potential energy of the system is obtained, then the Lagrangian can be found. A number of derivatives are then computed to yield the EOMs. The resultant nonlinear EOM are:

$$\left(J_r + J_p\, sin\,\alpha^2\right)\ddot{\theta} + m_p l r\, cos\,\alpha\ddot{\alpha} + 2J_p \sin\alpha\cos\alpha\dot{\theta}\dot{\alpha}$$
$$-m_p l r\sin\alpha\dot{\alpha}^2 = \tau - b_r \qquad (4)$$

and

$$J_p\ddot{\alpha} + m_p l r\, cos\,\alpha\,\ddot{\theta} - J_p\sin\alpha\cos\alpha\,\dot{\theta}^2 + m_p g l\sin\alpha = \qquad -b_p\dot{\alpha}$$
(5)

where $J_r = m_r r^2/3$ is the moment of inertia of the rotary arm with respect to the pivot (i.e. rotary arm axis of rotation) and $J_p = m_p L_p^2/3$ is the moment of inertia of the pendulum link relative to the pendulum pivot (i.e. axis of rotation of pendulum). The viscous damping acting on the rotary arm and the pendulum link are b_r and b_p, respectively. The applied torque at the base of the rotary arm generated by the servo motor is

$$\tau = \frac{k_m}{R_m}\left(v_m - k_m\dot{\theta}\right) \qquad (6)$$

When the nonlinear EOM are linearized about the operating point, the resultant linear EOM for the rotary pendulum is defined as:

$$J_r\ddot{\theta} + m_p lr\ddot{\alpha} = \tau - b_r\dot{\theta} \qquad (7)$$

and

$$J_p\ddot{\alpha} + m_p lr\ddot{\theta} + m_p gl\alpha = -b_p\dot{\alpha} \qquad (8)$$

Solving for the acceleration terms yields:

$$\ddot{\theta} = \frac{1}{J_t}(m_p^2 l^2 rg\alpha - J_p b_r\dot{\theta} + m_p lrb_p\dot{\alpha} + J_p\tau) \qquad (9)$$

and

$$\ddot{\alpha} = \frac{1}{J_t}(-m_p glJ_r\alpha + m_p lrb_r\dot{\theta} - J_p b_p\dot{\alpha} - m_p rl\tau) \qquad (10)$$

Where,

$$J_t = J_p J_r - m_p^2 l^2 r^2 \qquad (11)$$

The linear state space equations are

$$\dot{x}(t)=Ax(t)+Bu(t) \qquad (12)$$

and

$$y(t) = Cx(t) + Du(t) \qquad (13)$$

where *x* is the vector of state variables (*n x* 1), *u* is the control input vector (*r x* 1), *y* is the output vector (*m x* 1), *A* is the system matrix (*n x n*), *B* is the input matrix (*n x r*), *C* is the output matrix (*m x n*) and *D* is the feedforward matrix
(*m x r*).

For the rotary pendulum system, the state and output are defined

$$x(t) = [\theta(t) \;\; \alpha(t) \;\; \dot{\theta}(t) \;\; \dot{\alpha}(t)\,]^{\mathrm{T}} \qquad (14)$$

and

$$y(t) = [\theta(t) \;\; \alpha(t)]^{\mathrm{T}} \qquad (15)$$

Thus, the state space model obtained is

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 152.0057 & -10.1381 & -0.5005 \\ 0 & 264.3080 & -10.0202 & -0.8702 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 50.6372 \\ 50.0484 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

E. Swing-Up Control

In theory, if the arm angle is kept constant and the pendulum is given an initial perturbation, the pendulum will keep on swinging with constant amplitude. The idea of energy control is based on the preservation of energy in ideal systems: The sum of kinetic and potential energy is constant. However, friction will be damping the oscillation in practice and the overall system energy will not be constant. It is possible to capture the loss of energy with respect to the pivot acceleration, which in turn can be used to find a controller to swing up the pendulum. The nonlinear equation of motion of a single pendulum based on the diagram in figure 5 is

$$J_p\ddot{\alpha}(t) + m_pglsin\alpha(t) + m_plu(t)cos\alpha(t) = 0 \quad (16)$$

where $\alpha(t)$ is the angle of the pendulum defined as positive when rotated counter clockwise, J_p is the moment of inertia with respect to the pivot point, m_p is the mass of the pendulum link, l is the distance between the pivot and the center of mass, and $u(t)$ is the linear acceleration of the pendulum pivot (positive along the x_0 axis).

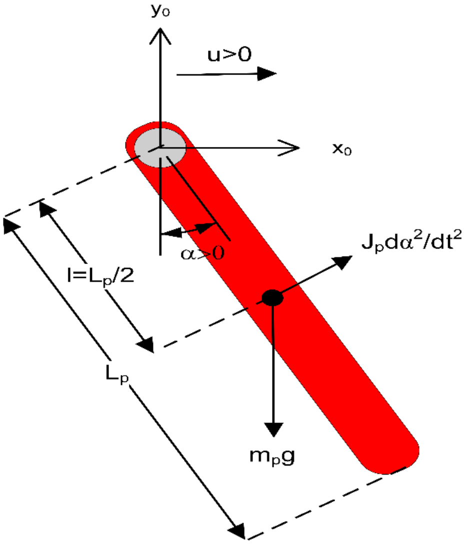


Fig. 5 Freebody Diagram of Pendulum

The potential energy of the pendulum is

$$E_p(t) = m_p gl(1 - cos\alpha)$$

and the kinetic energy is

$$E_k = \frac{1}{2} J_p \dot{\alpha}^2$$

Note that the moment of inertia used to define the pendulum kinetic energy is with respect to its center of mass. The potential energy is zero when the pendulum is at rest at $\alpha = 0$ and equals $E_p = 2\ m_p gl$ when the pendulum is upright at $\alpha = \pm\ \pi$. The sum of the potential and kinetic energy of the pendulum is

$$E = \frac{1}{2} J_p \dot{\alpha}^2 + m_p gl(1 - cos\alpha) \qquad (17)$$

Differentiating equation 17 yields,

$$\dot{E} = \frac{dE}{dt} = J_p \ddot{\alpha}\dot{\alpha} + m_p glsin\alpha\dot{\alpha}$$

(18)

Solving for $J_p \ddot{\alpha}$ in equation 16,

$$J_p \ddot{\alpha} = -m_p glsin\alpha - m_p ulcos\alpha$$

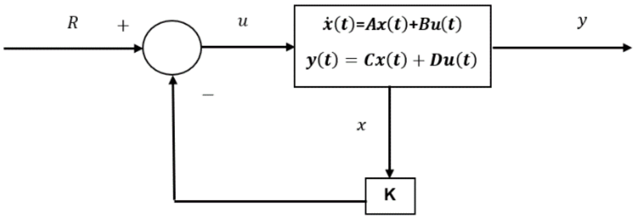


Fig. 6 LQR Block Diagram

and substituting this into equation 18 gives,

$$\dot{E} = -m_p ul\dot{\alpha}cos\alpha$$

Since the acceleration of the pivot is proportional to current driving the arm motor and thus also proportional to the motor voltage, it is possible to control the energy of the pendulum with the proportional control law

$$u = (E - E_r)\dot{\alpha}cos\alpha \qquad (19)$$

This control law will drive the energy of the pendulum towards the reference energy, i.e. $E(t) \rightarrow E_r$. By setting the reference energy to the pendulum potential energy, $E_r = E_p$, the control law will swing the link to its upright position. Notice that the control law is nonlinear because it includes nonlinear terms (e.g. $cos\alpha$). Further, the control changes sign when $\dot{\alpha}$ changes sign and when the angle is ± 90 degrees. For the system energy to change quickly, the magnitude of the control signal must be large. As a result, the following swing up controller is implemented in the controller as

$$u =\ sat_{\ u_{max}} (k_e(E - E_r)\text{sign}(\dot{\alpha}cos\alpha)) \qquad (20)$$

where k_e is a tunable control gain and the $sat_{\ u_{max}}$ function saturates the control signal at the maximum acceleration of the pendulum pivot, u_{max}. The expression $\text{sign}(\dot{\alpha}cos\alpha)$ is used to enable faster control switching. The control law in equation 20 finds the linear acceleration needed to swing up the pendulum. Because the control variable in the QUBE-Servo 2 is motor voltage, $v_m(t)$, the acceleration needs to be converted into voltage. This can be done using the expression

$$v_m(t) = \frac{R_m r m_r}{k_t} u(t)$$

where R_m is the motor resistance, k_t is the current torque constant of the motor, r is the length of the rotary arm and m_r is the mass of the rotary arm [5].

III. LQR CONTROLLER AND ADP ALGORITHM

A. LQR Controller

The LQR controller is a well-known method that provides optimally controlled feedback gains to enable the closed-loop stable and high-performance design of systems. The block diagram of LQR controller is shown in figure 6.

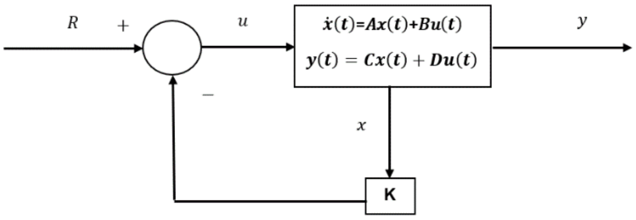


Fig. 6 LQR Block Diagram

LQR controller determines the feedback law to minimize the size of the state vector in the least time with the least control effort. The assumptions that are made while designing LQR controller are all the states of system are well known and the system is completely controllable. The settings of a LQR controller governing either a machine or process (like an airplane or chemical reactor) are found by using a mathematical algorithm that minimizes a cost function with weighting factors supplied by a human (engineer). The cost function is often defined as a sum of the deviations of key measurements, like altitude or process temperature, from their desired values and it is given by,

$$r(x,u) = \int_0^\infty (y^T(t)Q_y y(t) + u^T(t)Ru(t))dt \qquad (21)$$

Where $Q_y \in R^{p\times p} \geq 0$ and $R \in R^{m\times m} > 0$ are the weight matrices that are user-prescribed. The role of the weighting matrices Q and R is to establish a trade-off between performance and actuator effort. This Q weighting matrix refers to the performance and R weighting matrix refers to actuator effort [6].

B. ADP Algorithm

RL is a very useful tool in solving optimization problems by employing the principle of optimality from DP. In particular, in control systems community, RL is an important approach to handle optimal control problems for unknown nonlinear systems. DP provides an essential foundation for understanding RL. One class of RL methods is built upon the actor-critic structure, namely adaptive critic designs, where an actor component applies an action or control policy to the environment, and a critic component assesses the value of that action and the state resulting from it. The combination of DP, NN, and actor-critic structure results in the ADP algorithms [7].

There are many schemes available in ADP to enhance the LQR controller performance. Some of the algorithms are the model based iterative scheme, model free iterative scheme, dynamic output feedback scheme. The model based iterative scheme requires system dynamics for producing the output, whereas the model free iterative scheme continuously monitors the state trajectories of the system for producing the output and it also does not require the system dynamics [4].

The model based iterative scheme has two types which are Policy Iteration (PI) and Value Iteration (VI). The PI algorithm requires an initially stabilizing policy K_o and utilizes Lyapunov equation which makes computation process easier. The VI algorithm will perform recursive updates on the cost matrix P_i instead of solving the Lyapunov equation in every iteration. It no longer requires stable initial policy and generally take more iterations to converge. However, both algorithms are model-based as they require full model information (A, B, C) [4].

1) *Model Based Policy Iteration Algorithm*

It is one of the computational iterative methods. The key equation in this algorithm is the Lyapunov equation, which is easier to solve. This method essentially consists of a policy evaluation step followed by a policy update step. The first step in this algorithm is to compute the cost P_i of the control policy K_i by solving the Lyapunov equation 22. The second step is to compute an updated policy K_{i+1}. This PI algorithm requires an initially stabilizing policy K_o. For an open-loop stable system, the initial stabilizing policy K_o can be set to zero. However, for the case of unstable systems trial and error method should be followed for finding the initial stabilizing policy K_o in PI algorithm. The steps followed for obtaining the optimized K matrix are given below,

i. Initialize a stable control policy K_0.

ii. Evaluate Policy:

$$(A + BK_i)^T P_i + P_i\ (A + BK_i) + Q + {K_i}^T RK_i = 0$$

$$(A + BK_i)^T P_i\ + P_i(A + BK_i) = -(Q + {K_i}^T RK_i) \quad (22)$$

(A1)P + P(B1) = C1 which is similar to Sylvester equation which solves P for given A1,B1 and C1.

Where,

$$\text{A1} = (A + BK_i)^T$$

$$\text{B1} = A + BK_i$$

$$\text{C1} = -\left(Q + {K_i}^T RK_i\right)$$

iii. Improve Policy:

$$K_{i+1} = -R^{-1}B^T P_i \qquad (23)$$

iv. Repeat and Terminate:

Repeat with i = i+1 until

$$\|\mathrm{P_i} - \mathrm{P_{i-1}}\| < e$$

for some very small positive constant e [4].

IV. SOFTWARE IMPLEMENTATION

A. State Feedback Model

Initially, the system is unstable. In order to make the system stable, state feedback gain K is designed. The figure 7 shows the state feedback model of QUBE servo-2 plant.

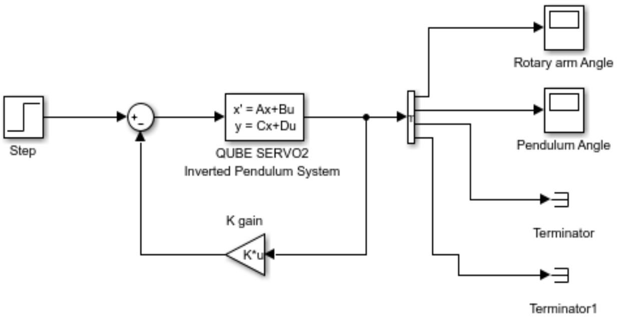


Fig. 7 State Feedback Simulink Block Diagram

In the above figure 7, state feedback gain K determines the stability of the plant. K gain is obtained using 2 different methods (LQR and ADP) and their performances were plotted and compared in the section V.

B. Swing-Up and Balance Control

The swing-up control is used for bringing the pendulum from downward position to upright position and the balance control is used for maintaining the pendulum at upright position within a tolerance limit. The swing-up and balance control implemented for rotary inverted pendulum system in MATLAB-SIMULINK platform is shown in figure 8.

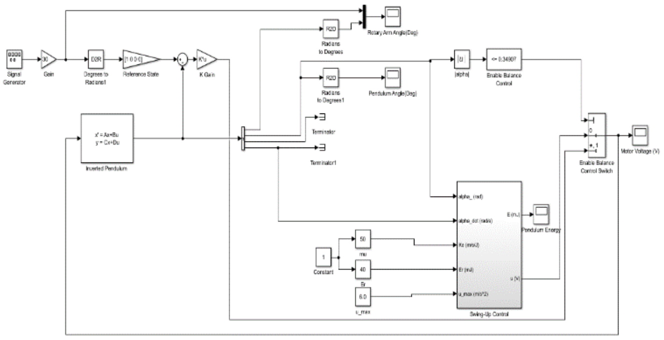


Fig. 8 Swing-Up and Balance Control Simulink Block Diagram

The swing-up control sub-block is shown in figure 9. The energy-based swing-up control is a sub-block in swing-up control block.

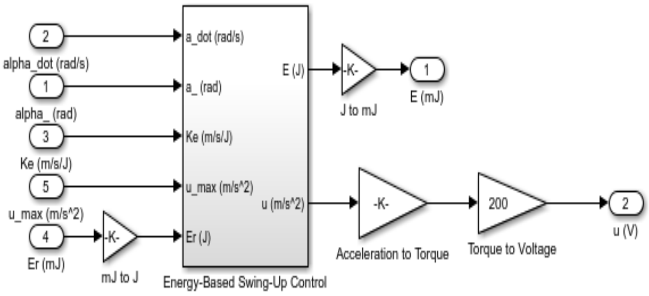


Fig. 9 Swing-up Control Sub Block

Energy-based swing-up control block shown in figure 10 will perform mathematical operations for the given inputs and provide outputs namely pendulum energy and linear acceleration of pendulum pivot. This control block is implemented based on the equation 20 discussed in section II.

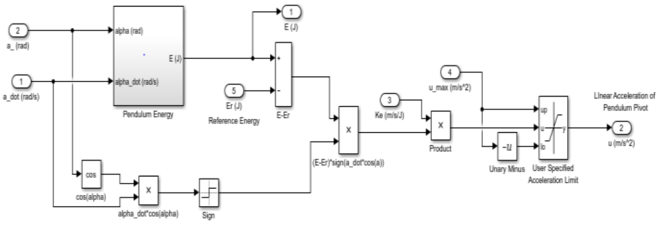


Fig. 10 Energy- Based Swing-Up Control Sub-Block

Last sub-block is pendulum energy which will give pendulum energy as output when the pendulum angle and its derivative are given as inputs. By using many sub-blocks, mathematical operations are performed to obtain swing-upcontrol which will bring up the pendulum to upright position.

V. RESULTS

A. Open Loop and Closed Loop Response

The open loop and closed loop response of the QUBE-Servo 2 rotary inverted pendulum system is shown in figure 11.

The QUBE-Servo 2 rotary inverted pendulum system is inherently open loop unstable and non-linear. As seen in the left side of the figure 11, it is evident that the pendulum angle is unstable and produces unbounded output. The optimal control is used to find the optimum controller gain to balance the pendulum at the upright position. LQR Controller is the optimal control used to determine the controller gain matrix K to make the inverted pendulum system closed loop stable. In the right side of figure 11, the closed loop response of pendulum angle is plotted. The K matrix (**K = [-1.0000 35.0244 -1.4474 3.0909]**) obtained by LQR Control is used in closed loop Simulink model shown in figure 7 to obtain the closed loop stable response. LQR controller based output have smoother performance, less setting time and the overshoot depends on Q and R Cost matrices.

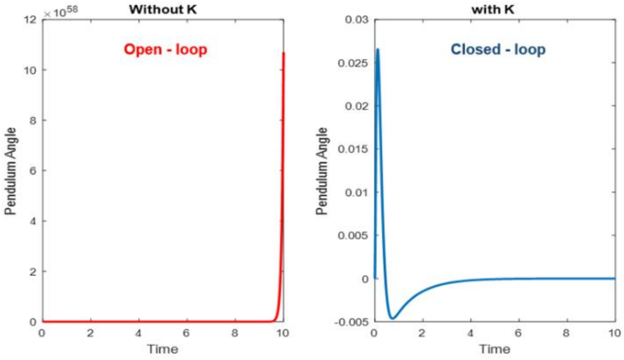


Fig. 11 Open Loop and Closed Loop Response

B. LQR BASED CONTROL OUTPUT

LQR based feedback gain K is determined using the MATLAB code (Refer Appendix) and its response is plotted in figure 12. By tuning Q and R values, the output will change. In the figure 12, two different Q matrices are used to find the gain matrix K. The response of pendulum angle for two different K gain matrices are plotted in figure 12.

R value is set to 1. The two different Q matrices are used to obtain two different outputs and those matrices are shown below. For Low Q matrix, the settling time and overshoot will be high. To obtain less settling time and minimum overshoot, High Q matrix can be used. Higher the Q value, better the performance. The comparative graph has been shown in the figure 12.

$$\text{Low Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{High Q} = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

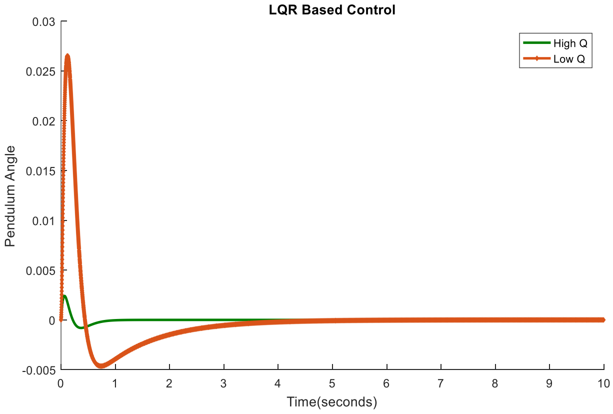


Fig. 12 LQR Based Control Response

C. ADP Based Control Output

By using the algorithm shown in section III, gain matrix K is obtained. Since it is an Iterative method and has a terminate condition (**‘e’ value**), improvement in each iteration and effectiveness of ‘e’ value has to be tested before implementing the value in the QUBE Servo-2 rotary inverted pendulum plant.

The figure 13 shows the K value based on different iterations and the figure 14 shows how the final K value of the algorithm changes based on ‘e’ value.

The figure 15 compares the K value of LQR and ADP. Both outputs will be similar since Q and R value are same for both methods. In LQR, A, B, Q and R matrices are required to find K matrix. In ADP, in addition to A, B, Q, R, the initial stabilizing gain matrix K0 is also needed such that the algorithm will improve the K value for each iteration and provides best possible K value based on given inputs and terminate condition.

Initial stabilizing gain matrix K0 (**K = [-2 80 -4 40]**) has worst performance as it has more oscillations and settles very slowly. The gain matrix K1 (**K1 = [-1.25 55.980906 -1.72095 20.3323283])** obtained after first iteration has less oscillations but it settles slowly. The gain matrix K2 (**K2 = [-1.000304 40.422411 -1.221734 6.028915]**) obtained after second iteration has no oscillations and settles quickly. The gain matrix K (**K = [-1.0000 20.6532 -1.0202 2.3935]**) obtained after the last iteration K has better performance and settles quickly. Thus, if the initial stabilizing gain matrix K0 is known, then the ADP Algorithm will evaluate the policy and produces better gain matrix which have less settling time and gives better performance compared to that of initial stabilizing matrix K0.

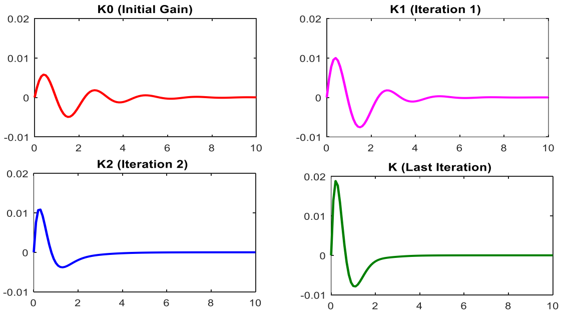


Fig. 13 Simulation Results of ADP Algorithm Based on Iteration

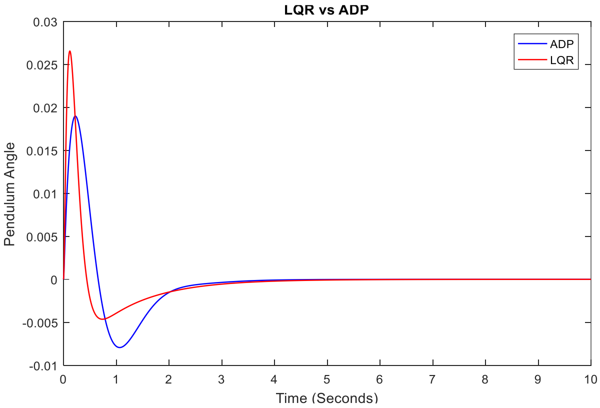
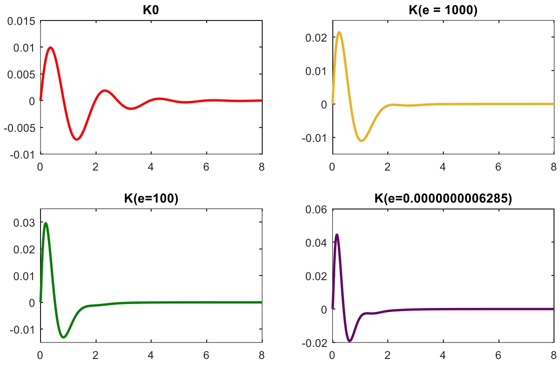


Fig. 14 Simulation Results of ADP Algorithm Based on ‘E’ Value to Terminate the Loop

The figure 14 will explain the importance of ‘*e*’ value in the elimination step of ADP algorithm. If the *e* value is 1000, the settling time is not very low. In the below figure, it is inferred that if the *e* value is reduced, the settling time will also reduce. In the ADP algorithm, *e* value should be a positive constant and less than 1, so the range of *e* is 0 to 1. Lesser the *e* value, smaller the settling time. For the QUBE-Servo 2 rotary inverted pendulum system, the smallest *e* value will be 0.000000000629, any value less than this smallest value has no impact on the K value. In the figure, the pendulum angle for the smallest *e* value has very low settling time and quicker response.



The figure 15 shows the comparison of conventional LQR and ADP based LQR controller pendulum angle output of QUBE-Servo 2 rotary inverted pendulum system. The LQR controller has better performance for the system with gain matrix **K = [-1.0000 35.0244 -1.4474 3.0909]**. In ADP algorithm, this gain is used as initial stabilizing gain matrix. The updated K matrix obtained from ADP algorithm **K = [-1.0000 20.6532 -1.0202 2.3935]** also gives better performance similar to LQR based control response. The only difference is that there will be a slight change in overshoot and settling time. This change has been plotted in the figure 15.

D. Swing-Up and Balance Control

The state feedback gain matrix K value obtained from LQR and ADP methods are used in swing-up and balance control Simulink block shown in figure 8 and their performances are compared here.

1) Initial Stabilizing K Matrix Response

The output for initial stabilizing K matrix used in ADP algorithm(**K = [-2 80 -4 40]**) is shown in figure 16, 17 and 18.

In figure 16, the actual rotary arm angle takes more seconds to track the set point rotary arm angle. In figure 17, the pendulum angle output has worst performance, more oscillations and very slow response. As this control gain has worst performance, it takes more pendulum energy to make the system stable. This control gain is given as initial stabilizing gain in the ADP algorithm to increase the performance. The updated K matrix from ADP algorithm and its output performances were shown in the figures 19, 20 and 21. The outputs for initial stabilizing K matrix are shown in below figures 16 to 18.

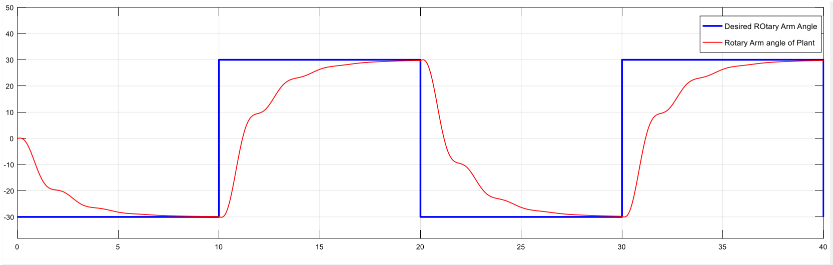


Fig. 16 Desired vs System Rotary Arm Angle for Initial Stabilizing K Matrix Used in ADP algorithm

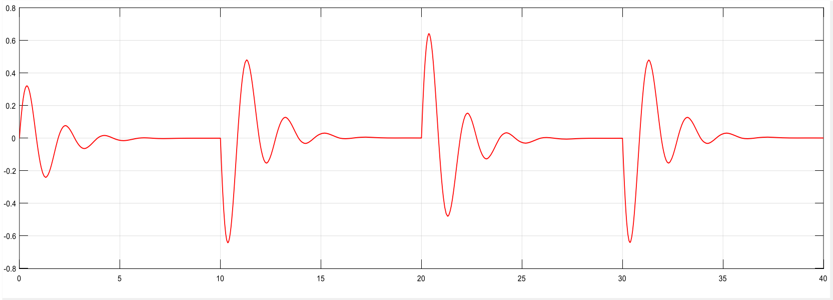


Fig. 17 Pendulum Angle for Initial Stabilizing K Matrix Used in ADP Algorithm

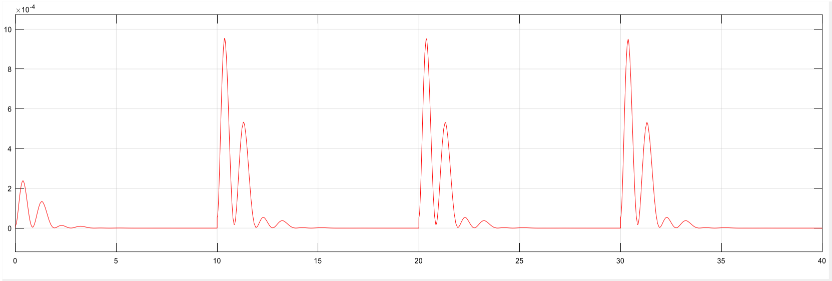


Fig. 18 Pendulum Energy for Initial Stabilizing K Matrix Used in ADP Algorithm

From figure 16, 17, 18, it is inferred that the initial stabilizing K matrix produces worst performance with more settling time and oscillations. Here the pendulum takes around 7 seconds to settle and also it consumes more energy for settling. Thus, using this K matrix value for controlling the rotary inverted pendulum system is not advisable.

2) ADP Based K Matrix Response

The output for updated K matrix (**K = [-1.0000 20.6532 -1.0202 2.3935]**) obtained by ADP algorithm is shown in figure 19, 20 and 21.

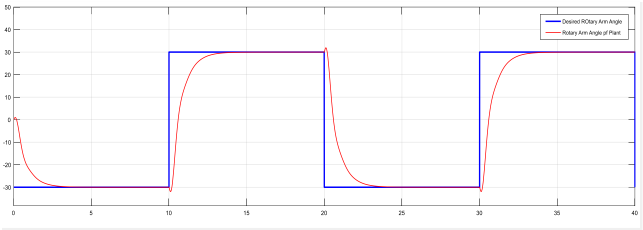


Fig. 19 Desired vs System Rotary Arm Angle for K Matrix Updated by ADP

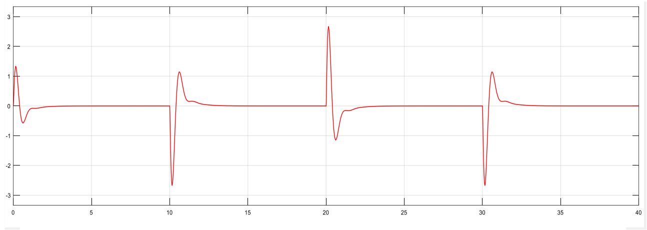


Fig. 20 Pendulum Angle for K Matrix Updated by ADP

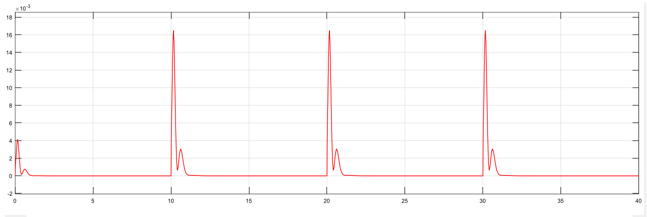


Fig. 21 Pendulum Energy for K Matrix Updated by ADP

From figure 19, 20, 21, it is inferred that the ADP based K matrix produces best performance with less settling but with more overshoot. It also consumes less pendulum energy to settle the pendulum at the right position. When compared with initial stabilizing K matrix response, the updated K matrix obtained from ADP algorithm has smooth performance and quicker response but with slightly more overshoot. Thus, using the K matrix obtained from ADP algorithm, it is possible to obtain best performance results for QUBE-Servo 2 rotary inverted pendulum system.

3) LQR Based K Matrix Response

The output for updated K matrix (**K = [-1.0000 35.0244 -1.4474 3.0909]**) obtained by LQR is shown in figure 22, 23 and 24.

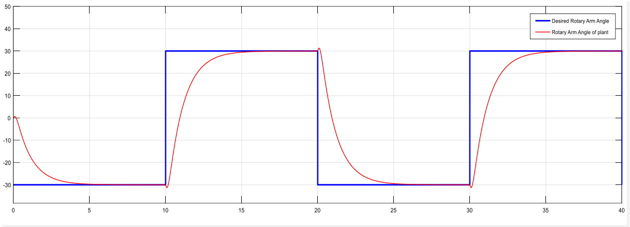


Fig. 22 Desired vs System Rotary Arm Angle for K Matrix Obtained by LQR

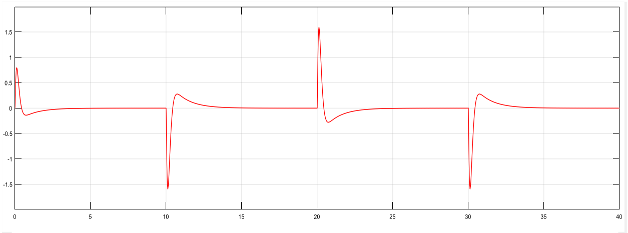


Fig. 23 Pendulum Angle for K Matrix Obtained by LQR

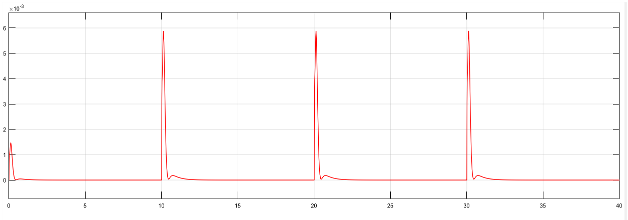


Fig. 24 Pendulum Energy for K Matrix Obtained by LQR

The outputs obtained using LQR gain has similar performance when compared to updated K matrix obtained from ADP algorithm. ADP based output has less settling time and slightly more overshoot when compared with LQR based output. From the figures 16 to 24, it is inferred that the initial stabilizing K matrix used in ADP algorithm has worst performance compared to other two K matrix values.

The rotary arm angle of the plant will track the desired angle very slowly in figure 16 (initial stabilizing K matrix), whereas it tracks very quickly in figure 19 (updated ADP K matrix). The performance of LQR based K matrix is little slower than the updated ADP K matrix.

Comparing the pendulum angle responses for three different K matrix, it is confirmed that the updated K matrix obtained by ADP algorithm settles very quickly but has higher peak overshoot when compared with the responses obtained from LQR based K matrix.

VI. Conclusion

Thus, the ADP algorithm is trained effectively using MATLAB-SIMULINK and the updated gain matrix K is used to obtain balance control. The simulation results presented shows that the ADP based LQR control gives better performance and the output settles little quickly than conventional LQR controller. The future scope is to implement the ADP based LQR controller in real time system to achieve better performance and provide real time disturbances to analyse the efficiency of ADP based LQR controller in real time.

REFERENCES

[1] H. Wang, H. Dong, L. He, Y. Shi and Y. Zhang, "Design and Simulation of LQR Controller with the Linear Inverted Pendulum," *2010 International Conference on Electrical and Control Engineering*, 2010, pp. 699-702.

[2] F. A. Yaghmaie and S. Gunnarsson, "A New Result on Robust Adaptive Dynamic Programming for Uncertain Partially Linear Systems," *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 7480-7485.

[3] Y. Liu, Y. Luo and H. Zhang, "Adaptive dynamic programming for discrete-time LQR optimal tracking control problems with unknown dynamics," *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2014, pp. 1-6.

[4] S. A. A. Rizvi and Z. Lin, "Reinforcement Learning-Based Linear Quadratic Regulation of Continuous-Time Systems Using Dynamic Output Feedback," in *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4670-4679, Nov. 2020..

[5] https://www.quanser.com/products/qube-servo-2/

[6] https://en.wikipedia.org/wiki/Linear%E2%80%93quadratic_regulator

[7] Huaguang Zhang, Derong Liu, Yanhong Luo, Ding Wang, “Adaptive Dynamic Programming for Control”, Springer, Switzerland, 2013.