OPTIMAL LQR BALANCE CONTROL OF PENDULUM

Topics Covered

- Stability analysis.
- State-feedback control.
- Linear Quadratic Regulator (LQR) optimization.

Prerequisites

- Filtering laboratory experiment.
- Rotary Pendulum Modeling laboratory experiment.

1 Background

Linear Quadratic Regulator (LQR) theory is a technique that is ideally suited for finding the parameters of the pendulum balance controller (e.g. in the Balance Control laboratory experiment).

Given that the equations of motion of the system can be described in the form

The standard state-space representation of a multi-input multi-output (MIMO) continuous linear-time invariant (LTI) system with n state variables, r input variables, and m output variables is

$$\dot{x}(t) = Ax + Bu \tag{1.1}$$

and

$$y(t) = Cx(t) + Du(t) \tag{1.2}$$

where $x \in \Re^{n\times 1}$ is the vector of state variables, $u \in \Re^{r\times 1}$ is the control input vector, $y \in \Re^{m\times 1}$ is the output vector, $A \in \Re^{n\times n}$ is the system matrix, $B \in \Re^{n\times r}$ is the input matrix, $C \in \Re^{m\times n}$ is the output matrix, and $D \in \Re^{m\times r}$ is the feed-forward matrix.

The state-feedback control of a system is shown in Figure 1.1. Let u be a state feedback control law of the form

$$u = -Kx \tag{1.3}$$

where $K \in \Re^{m\times n}$ is the feedback gain. Applying this to Equation 1.1 gives the closed-loop system

$$\dot{x} = Ax - BKx = (A - BK)x. \tag{1.4}$$

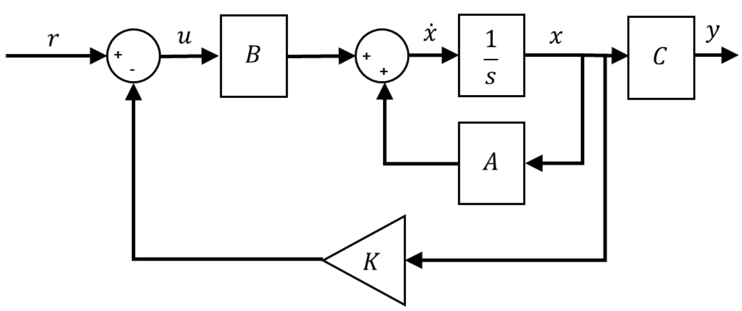


Figure 1.1: State feedback block diagram

The LQR algorithm computes a control law u such that the performance criterion or cost function

$$J = \int_0^{\infty} \left(x_{ref} - x(t)\right)^T Q \left(x_{ref} - x(t)\right) + u(t)^T R u(t) dt \tag{1.5}$$

is minimized. The design matrices Q and R hold the penalties on the deviations of state variables from their setpoint and the control actions, respectively. When an element of Q is increased, therefore, the cost function increases the penalty associated with any deviations from the desired setpoint of that state variable, and thus the specific control gain will be larger. When the values of the R matrix are increased, a larger penalty is applied to the aggressiveness of the control action, and the control gains are uniformly decreased.

The state vector x of the rotary pendulum system is defined

$$x = \begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} \end{bmatrix}^\top . \tag{1.6}$$

The reference signal includes the desired rotary arm position, θ_r and is defined

$$x_{ref} = \begin{bmatrix} \theta_r & 0 & 0 & 0 \end{bmatrix}^\top \tag{1.7}$$

and the control strategy used to balance the pendulum (and track the rotary arm setpoint) is

$$u = K(x_{ref} - x) = k_{p,\theta}(\theta_r - \theta) - k_{p,\alpha}\alpha - k_{d,\theta}\dot{\theta} - k_{d,\alpha}\dot{\alpha}. \tag{1.8}$$

This control law is a state-feedback control and is illustrated in Figure 1.1. The structure is equivalent to the PD control used to balance the pendulum in the Balance Control laboratory experiment.

2 In-Lab Exercises

Construct a QUARC® controller similarly to Figure 2.1 that balances the pendulum on the QUBE-Servo 2 rotary pendulum system using a generated control gain K.

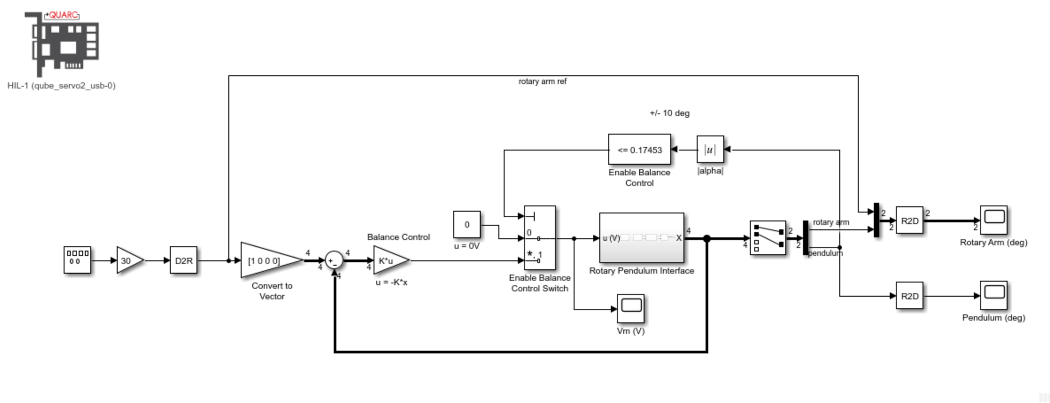


Figure 2.1: SIMULINK® model used with QUARC® run optimized balance controller

The LQR theory has been packaged in the MATLAB® *Control System Toolbox*. Given the model of the system,in the form of the state-space matrices A and B, and the weighting matrices Q and R, the LQR function computes the feedback control gain automatically.

In this experiment, the state-space model is already available. In the laboratory, the effect of changing the Q weighting matrix while R is fixed to 1 on the cost function J will be explored.

2.1 LQR Control Design

1. In MATLAB®, run the `setup_qube2_rotpen.m` script. This loads the QUBE-Servo 2 rotary pendulum state-space model matrices A, B, C, and D. The A and B matrices should be displayed in the Command Window:

```
A =

         0         0    1.0000         0
         0         0         0    1.0000
         0  152.0057  -12.2542   -0.5005
         0  264.3080  -12.1117   -0.5064


B =

         0
         0
   50.6372
   50.0484
```

2. Use the `eig` command to find the open-loop poles of the system. What do you notice about the location of the open-loop poles? How does that affect the system?

3. Using the `lqr` function with the loaded model and the weighting matrices

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad R = 1,$$

   generate gain K. Give the value of the control gain generated.

4. Change the LQR weighting matrix to the following and generate a new gain control gain:

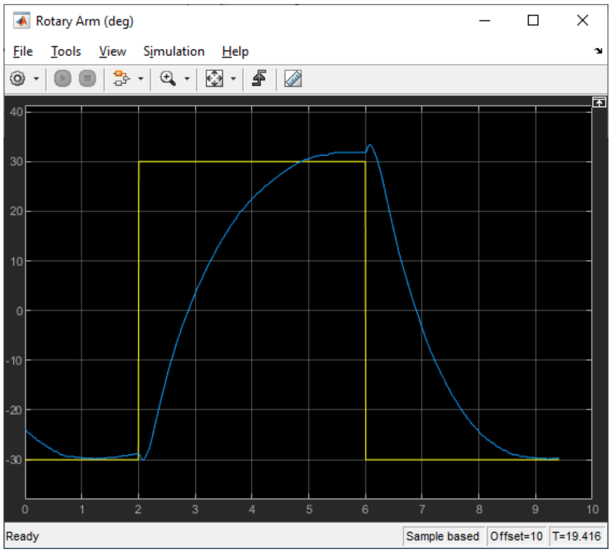
$$Q = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad R = 1.$$

   Record the gain generated. How does changing q_{11} affect the generated control gain? Based on the description of LQR in 2.1, is this what you expected?

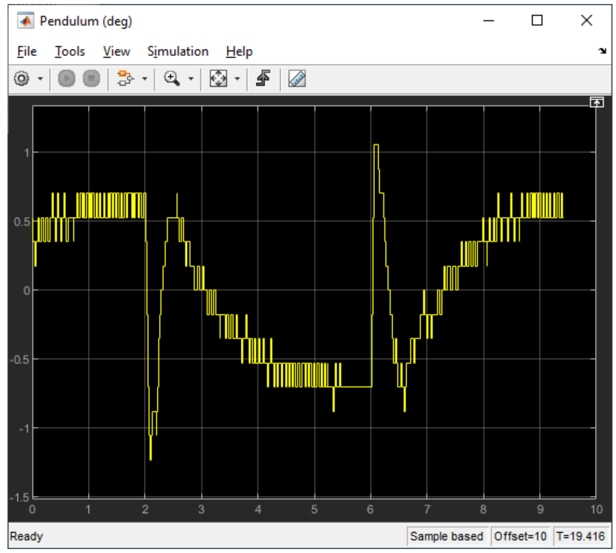
2.2 LQR-Based Balance Control

1. Run the `setup_qube2_rotpen.m` script in MATLAB.
2. Using the SIMULINK® model you made in the Rotary Pendulum Modeling laboratory experiment, construct the controller shown in Figure 2.1:
   - As done in the Rotary Pendulum Modeling laboratory experiment, use gain blocks used to convert encoder counts to angles
   - Build state x given in Equation 1.6. Use the Transfer Fcn Simulink blocks with $50s/(s+50)$ to compute the velocities $\dot{\theta}$ and $\dot{\alpha}$. Recall that this takes the derivative and filters out the noise using a low-pass filter.
   - Add the necessary Sum and Gain blocks to implement the state-feedback control given in Equation 1.8. Since the control gain is a vector, make sure the gain block is configured to do matrix type multiplication.
   - Add the Signal Generator block in order to generate a varying, desired arm angle. To generate a reference state, make sure you include a Gain block of $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$.
3. Load the gain designed in Step 3 in Section 2.1. Make sure it is set as variable `K` in the MATLAB® workspace.
4. Set the Signal Generator block to the following:
   - Type = Square
   - Amplitude = 1
   - Frequency = 0.125 Hz
5. Set the Gain block that is connected to the Signal Generator to 0.
6. Build and run the QUARC® controller.
7. Manually rotate the pendulum in the upright position until the controller engages.
8. Once the pendulum is balanced, set the Gain to 30 to make the arm angle go between $\pm 30°$. The scopes should read something similar as shown in Figure 2.2. Attach your response of the rotary arm, pendulum, and controller voltage.

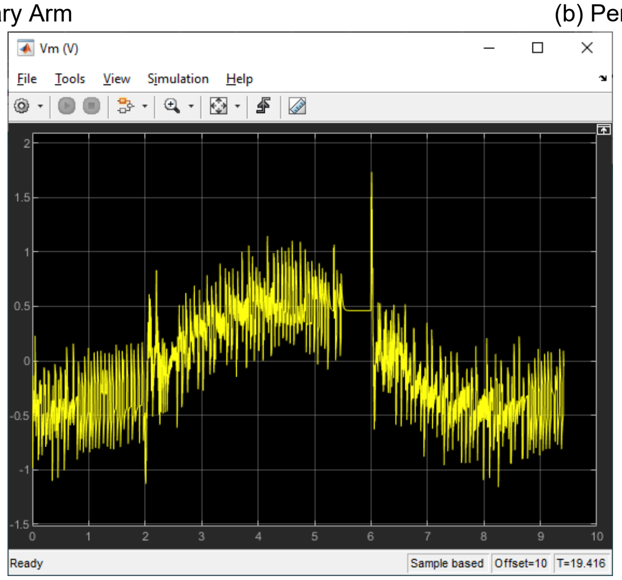
   Note: The rotary arm and pendulum response can vary depending on a number of conditions. Make sure the QUBE-Servo 2 is placed on a balanced surface and that the pendulum encoder cable is positioned to be directly overhead the pendulum encoder and rotary arm hub. This can minimize the amount of offset or error seen in the rotary arm response, for intance.



(a) Rotary Arm



(b) Pendulum



(c) Motor Voltage

Figure 2.2: QUBE-Servo 2 rotary pendulum response

9. In MATLAB®, generate the gain using $Q = \texttt{diag}(\begin{bmatrix} 5 & 1 & 1 & 1 \end{bmatrix})$ performed in Step 4 in the first part of this laboratory experiment. The `diag` command specifies the diagonal elements in a square matrix.
10. To apply the newly designed gain to the running QUARC controller, go to the Simulink model and select *Modeling* | *Update Diagram* or press CTRL-D on your keyboard.
11. Examine and describe the change in the *Rotary Arm (deg)* and *Pendulum (deg)* scopes.
12. Adjust the diagonal elements of Q matrix to reduce how much the pendulum angle deflects (overshoots) when the arm angle changes. Describe your experimental procedure to find the necessary control gain.
13. List the resulting LQR Q matrix and control gain K used to yield the desired results. Attach the responses using this new control gain and briefly outline how the response changed.

14. Stop the QUARC® controller.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com