ECE 147C Final Project

Max Crisafulli & Eric Hsieh

ECE 147C

June 5, 2023

Why Use This Hardware?

- Quanser Hardware allows for easy interaction with MATLAB/SIMULINK via the Quarc package
- Classic (and fun) toy example of LQR control of an unstable system.

Figure 1: The Quanser SRV02 Rotational Pendulum

Equations of Motion - Conventions

Lagrangian EoM Method

$$\mathcal{L} = T^* - V$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\xi}_i} - \frac{\partial \mathcal{L}}{\partial \xi_i} = \Xi_i$$

$$\xi_i = \begin{bmatrix} \theta \\ \alpha \end{bmatrix}$$

$$\Xi_i = \begin{bmatrix} \tau - B_r\dot{\theta} \\ -B_p\dot{\alpha} \end{bmatrix}$$

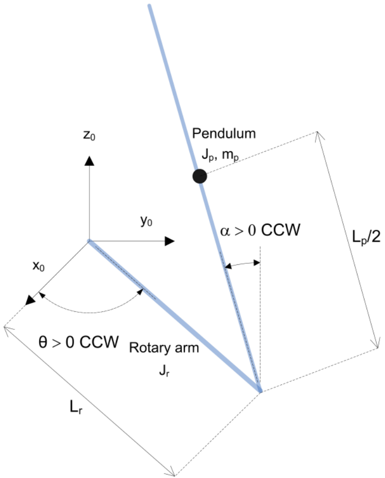

Figure 2: Inverted Pendulum Diagram

Equations of Motion - Full Nonlinear

EoM # 1 - $\xi_1 = \theta$: $\dfrac{d}{dt}\dfrac{\partial \mathcal{L}}{\partial \dot{\theta}} - \dfrac{\partial \mathcal{L}}{\partial \theta} = \tau - B_r\dot{\theta}$

$$\left(m_pL_r^2 + \frac{1}{4}m_pL_p^2 - \frac{1}{4}m_pL_p\cos(\alpha)^2 + J_r\right)\ddot{\theta} - \left(\frac{1}{2}m_pL_pL_r\cos(\alpha)\right)\ddot{\alpha}$$

$$+\left(\frac{1}{2}m_pL_p^2\sin(\alpha)\cos(\alpha)\right)\dot{\theta}\dot{\alpha} + \left(\frac{1}{2}m_pL_pL_r\sin(\alpha)\right)\dot{\alpha}^2 = \tau - B_r\dot{\theta}$$

EoM # 2 - $\xi_2 = \alpha$: $\dfrac{d}{dt}\dfrac{\partial \mathcal{L}}{\partial \dot{\alpha}} - \dfrac{\partial \mathcal{L}}{\partial \alpha} = -B_p\dot{\alpha}$

$$-\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta} + \left(J_p + \frac{1}{4}m_pL_p^2\right)\ddot{\alpha} - \frac{1}{4}m_pL_p^2\cos(\alpha)\sin(\alpha)\dot{\theta}^2$$

$$-\frac{1}{2}m_pL_pg\sin(\alpha) = -B_p\dot{\alpha}$$

Torque as a Function of Voltage: $\tau = \dfrac{\eta_gK_g\eta_mk_t(V_m - K_gk_m\dot{\theta})}{R_m}$

Equations of Motion - Linearization

With a non-linear $f(z)$, you can linearize around the upright position as follows:

$$z^T = [\theta, \alpha, \dot{\theta}, \dot{\alpha}, \ddot{\theta}, \ddot{\alpha}] \;\; \& \;\; z_0^T = [0,0,0,0,0,0] \;\; \text{(Upright Equilibrium)}$$

General Linearization Form: $f_{\text{lin}}(z) = f(z_0) + \sum_{\forall z} \left(\left. \frac{\partial f(z)}{\partial z_i} \right|_{z=z_0} \cdot z_i \right)$

Linearized EoM #1: $(m_p L_r^2 + J_r)\ddot{\theta} - \frac{1}{2} m_p L_p L_r \ddot{\alpha} = \tau - B_r \dot{\theta}$

Linearized EoM #2: $-\frac{1}{2} m_p L_p L_r \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2 \right) \ddot{\alpha} - \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha}$

Organizing into matrix EoM form $M(q)\ddot{q} + B(q,\dot{q})\dot{q} + G(q) = \tau, \;\; q^T = [\theta, \; \alpha]$

$$\begin{bmatrix} m_p L_r^2 + J_r & -\frac{1}{2} m_p L_p L_r \\ -\frac{1}{2} m_p L_p L_r & J_p + \frac{1}{4} m_p L_p^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} B_r & 0 \\ 0 & B_p \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{2} m_p L_p g \alpha \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

Reorganizing: $\begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \frac{1}{J_T} \begin{bmatrix} J_p + \frac{1}{4} m_p L_p^2 & \frac{1}{2} m_p L_p L_r \\ \frac{1}{2} m_p L_p L_r & J_r + m_p L_r^2 \end{bmatrix} \begin{bmatrix} \tau - B_r \dot{\theta} \\ \frac{1}{2} m_p L_p g \alpha - B_p \dot{\alpha} \end{bmatrix}$

Where $J_T = \det(M(q))$

Linearized Matrices (Continuous State Space Model)

$$\ddot{\theta} = \frac{1}{J_T}\left(-\left(J_p + \frac{1}{4}m_pL_p^2\right)B_r\dot{\theta} - \frac{1}{2}m_pL_pL_rB_p\dot{\alpha} + \frac{1}{4}m_p^2L_p^2L_rg\alpha + \left(J_p + \frac{1}{4}m_pL_p^2\right)\tau\right)$$

$$\ddot{\alpha} = \frac{1}{J_T}\left(\frac{1}{2}m_pL_pL_rB_r\dot{\theta} - (J_r + m_pL_r^2)B_p\dot{\alpha} + \frac{1}{2}m_pL_pg(J_r + m_pL_r^2)\alpha + \frac{1}{2}m_pL_pL_r\tau\right)$$

$$A = \frac{1}{J_T}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{4}m_p^2L_p^2L_rg & -(J_p + \frac{1}{4}m_pL_p^2)B_r & -\frac{1}{2}m_pL_pL_rB_p \\ 0 & \frac{1}{2}m_pL_pg(J_r + m_pL_r^2) & \frac{1}{2}m_pL_pL_rB_r & -(J_r + m_pL_r^2)B_p \end{bmatrix}$$

$$B = \frac{K_gk_t}{R_mJ_T}\begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_pL_p^2 \\ \frac{1}{2}m_pL_pL_r \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad x = \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix}$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 81.40 & -10.25 & -0.93 \\ 0 & 122.05 & -10.33 & -1.39 \end{bmatrix}\begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 83.47 \\ -80.32 \end{bmatrix} V_m$$

Energy-Based Swing Up Control

NOTE: The α used in this context is $\alpha = 0$ being the DOWNWARD position.

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

$$E_p = \frac{1}{2}m_p g L_p(1 - \cos(\alpha)),\ E_k = \frac{1}{2}J_p\dot{\alpha}^2,\ E = E_p + E_k$$

Using the idea of nonlinear control:

$$\tau = \frac{\eta_g K_g \eta_m k_t(u - K_g k_m \dot{\theta})}{R_m}$$

$$u = \frac{m_r L_r R_m}{\eta_g K_g \eta_m k_t} \cdot \mu \cdot \text{sign}\{\cos(\alpha)\dot{\alpha}\} \cdot (E - E_r) + K_g k_m \dot{\theta}$$

Where μ is the swing-up gain and E_r is the potential energy of the upright pendulum ($\alpha = \pi$), in this case $m_p g L_p = 0.42$J.

LQR Balancing Control was designed using Bryson's Rule where:

$$Q_{ii} = \frac{1}{(x_{i,max})^2}, \text{ and } R_{ii} = \frac{1}{(u_{max})^2}$$

$$x_{max} = \begin{bmatrix} \theta_{max} \\ \alpha_{max} \\ \dot{\theta}_{max} \\ \dot{\alpha}_{max} \end{bmatrix} = \begin{bmatrix} 4^o \cdot \frac{\pi}{180^o} \text{ rad} \\ 1.5^o \cdot \frac{\pi}{180^o} \text{ rad} \\ 100 \ \frac{\text{rad}}{\text{sec}} \\ 1 \ \frac{\text{rad}}{\text{sec}} \end{bmatrix} \ \& \ u_{max} = 5\text{V}$$

The states θ and α were measured and used to predict ALL the states using the linearized system model in a Kalman Filter (LQG) state estimator.

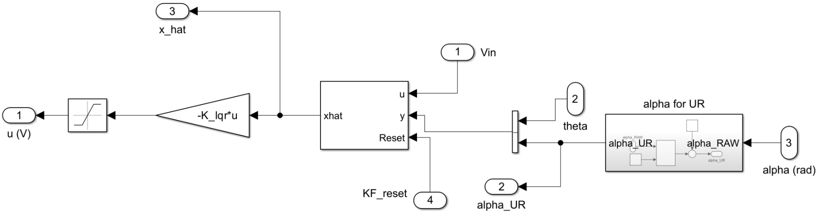

Figure 3: LQR Kalman Filter Feedback

Kalman Filter State Estimation

For a discrete-time linear system described by the following form. Our discrete SS model was generated via the `c2d` command in MATLAB from our continuous model. The A, B, C matrices in this case are the outputs of the `c2d` function.

$$x_{k+1} = A_k x_k + B_k u_k + w_k,\ w_k \sim \mathcal{N}(0,\ Q_k)$$

$$y_k = C_k x_k + v_k,\ v_k \sim \mathcal{N}(0,\ R_k)$$

The Kalman Filter Equations are defined as follows.

Predict State: $\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_k$

Predict Covariance: $\hat{P}_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$

Kalman Gain: $K_k = \hat{P}_{k|k-1} C_k^T (C_k \hat{P}_{k|k-1} C_k^T + R_k)^{-1}$

Update State: $x_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1})$

Update Covariance: $P_{k|k} = (I - K_k C_k) \hat{P}_{k|k-1}$

Switching Logic & Upright Angle

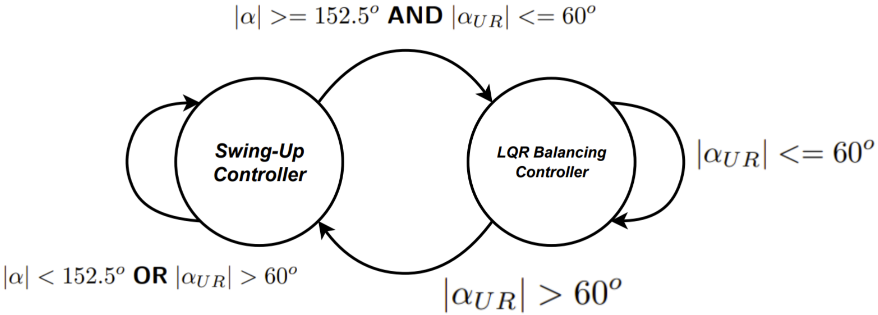

Figure 4: Controller Switching Logic

NOTE: In this case α is the ABSOLUTE sensor measurement where $\alpha = 0$ is the downward position. The upright alpha α_{UR} is the augmented α for use with the linearized LQR controller and Kalman Filter state estimator, defined as follows:

$$\alpha_{UR} = \mod(\alpha, 2\pi) - \pi$$

Implementation Results

Please turn your attention to the inverted pendulum for a demonstration...