ELEN90064
Advanced Control Systems

Phase 3 - Single-Link Flexible-Joint Robot

Xing Yang Goh
# 1001969
October 29, 2022

1 Introduction

Optimal control methods using a Linear Quadratic Regulator (LQR) controller and a Kalman Filter also known as a Linear Quadratic Estimation (LQE) will be combined to perform Linear Quadratic Gaussian (LQG) control on a single-link flexible-joint robot with motor 1 as the actuator shown in figure 1. The governing equations have been previously derived and expressed in equation 1, with the equilibrium point chosen at $x_{2e} = 45°$ such that the incremental variables will have a maximum deviation of ±15° since the controller steps up and down from 30° to 60° during operation. This report will cover the design and tuning of the LQG control using the separation principle to design the LQR controller and Kalman filter independently. Note that the analysis will be done in discrete time with a controller operating at 50Hz.

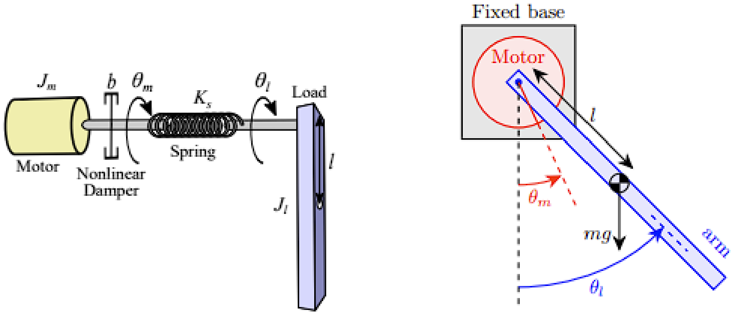


Figure 1: Schematic of a single-link flexible-joint robot

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} \theta_m \\ \theta_l \\ \dot{\theta}_m \\ \dot{\theta}_l \\ I_a \end{bmatrix} \quad ; \quad \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \\ \dot{\tilde{x}}_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{K_s}{J_m} & \frac{K_s}{J_m} & 0 & 0 & \frac{K_m}{J_m} \\ \frac{K_s}{J_l} & -\frac{K_s}{J_l} - \frac{mgl}{\sqrt{2}J_l} & 0 & 0 & 0 \\ 0 & 0 & -\frac{K_m}{L_a} & 0 & -\frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \\ \tilde{x}_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix} \tilde{u} \tag{1}$$

The specifications for the non-linear physical system operating between 30° and 60° are the following parameters:

1. Maximum overshoot of at most 5 degrees.
2. Settling time of at most 2.0 seconds (for a less than 2% settling criterion).
3. Steady state error of at most 2 degrees.
4. Maximum controller sampling frequency of 50 Hz.

2 Linear Quadratic Regulator Design

Before designing a full-state LQR controller, we have to calculate the controllability of the system, which is the rank of the matrix $\Gamma_c[\mathbf{A}, \mathbf{B}] \triangleq \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \mathbf{A}^3\mathbf{B} & \mathbf{A}^4\mathbf{B} \end{bmatrix}$. Using the A and B state space equation defined in equation 1, we can determine that the system is completely controllable since the row rank of the controllability matrix is 5.

2.1 Symmetric Root Locus

The symmetric root locus is a graphical design technique that identifies the following cost function:

$$J = \int_0^\infty \left[\rho |y_p(t)|^2 + |u(t)|^2 \right] dt \tag{2}$$

This ρ value will find the control law that minimises J with the linear full state feedback control $u = -\mathbf{Kx}$. To determine an appropriate ρ value, we will utilise second-order design principles as a guideline for pole placements. For a maximum overshoot of at most 5 degrees, we can estimate a minimum overshoot percentage M_p when the system moves 30°, which is the maximum step size for the system due to the nominal operation of the system between 30° and 60°. $M_p = \frac{5}{30} \times 100 = 16.67\%$. Using the relationship

$M_p = \exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right)$, we obtain $\zeta = 0.495$. Therefore, the bound for the complex domain angle beginning at the imaginary axis is:

$$\arcsin 0.495 = 29.7^\circ \tag{3}$$

For a setting time of at most 2 seconds, the 2% settling time criterion is $\sigma \geq \frac{4}{t_s}$, where sigma represents the distance between the imaginary axis and the pole boundary in the LHP.

$$\sigma \geq 2 \tag{4}$$

For the tracking of the pulse signals between 30° and 60°, this can be modelled as a sequence of step responses given settling time quicker than the period of the pulses. Therefore, we augment the discretised state space representation in equation 1 with an integral error state to allow the controller to converge to the reference signal with the internal model principle.

$$\begin{gathered}
\begin{bmatrix} x_I(k+1) \\ \mathbf{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{C} \\ \mathbf{0} & \mathbf{\Phi} \end{bmatrix} \begin{bmatrix} x_I(k) \\ \mathbf{x}(k) \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ \mathbf{0} & \mathbf{\Gamma} \end{bmatrix} \begin{bmatrix} r(k) \\ u(k) \end{bmatrix} \\
u(k) = -\begin{bmatrix} K_I & \mathbf{K} \end{bmatrix} \begin{bmatrix} x_I(k) \\ \mathbf{x}(k) \end{bmatrix}
\end{gathered} \tag{5}$$

To determine the symmetric root locus for analysis, we consider this augmented plant with integral control, identifying the SRL equation $1 + \rho G_0(-s)G_0(s) = 0$, where $G_0(s)$ is the open loop transfer function from the input voltage to θ_l:

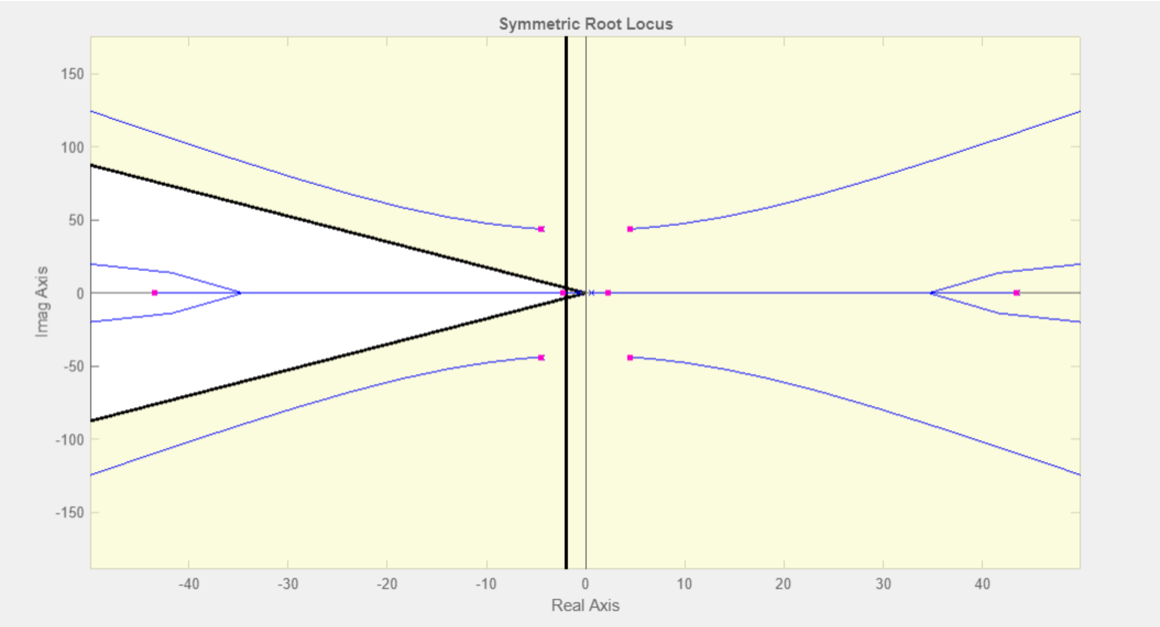


Figure 2: Symmetric Root Locus for θ_l with design constraints and $\rho = 1.5$

With the aforementioned second-order design constraints, a suitable value of $\rho = 1.5$ is chosen such that the poles of the system satisfies the settling time parameter from second-order design principles with $\sigma \geq 2$. Note that although the system is a higher order system, the pole-zero map of this SRL indicates that the system can be approximated as a third order transfer function, with the most dominant pole lying on the LHP of the real axis and a pair of complex conjugate poles. Although the designed overshoot parameter is unable to be adhered to due to the complex conjugate poles lying beyond the 29.7° range for any ρ value, this overshoot is reduced due to the presence of the most dominant pole on the real axis since these second order design principles are rough guidelines, especially since our system is a third-order system which behaves differently. Note that the poles on the right-half plane can be ignored in the SRL pole-zero map and do not represent unstable poles in the system. Since the LQR controller cost function can be expressed as:

$$J = \sum_{k=0}^{\infty} \left[x^T[k]Qx[k] + u^T[k]Ru[k]\right] \tag{6}$$

We can define the following parameters for the Q and R optimisation matrices by comparing terms in equation 2 and using a full state observer assumption with $C = \mathbb{1}$:

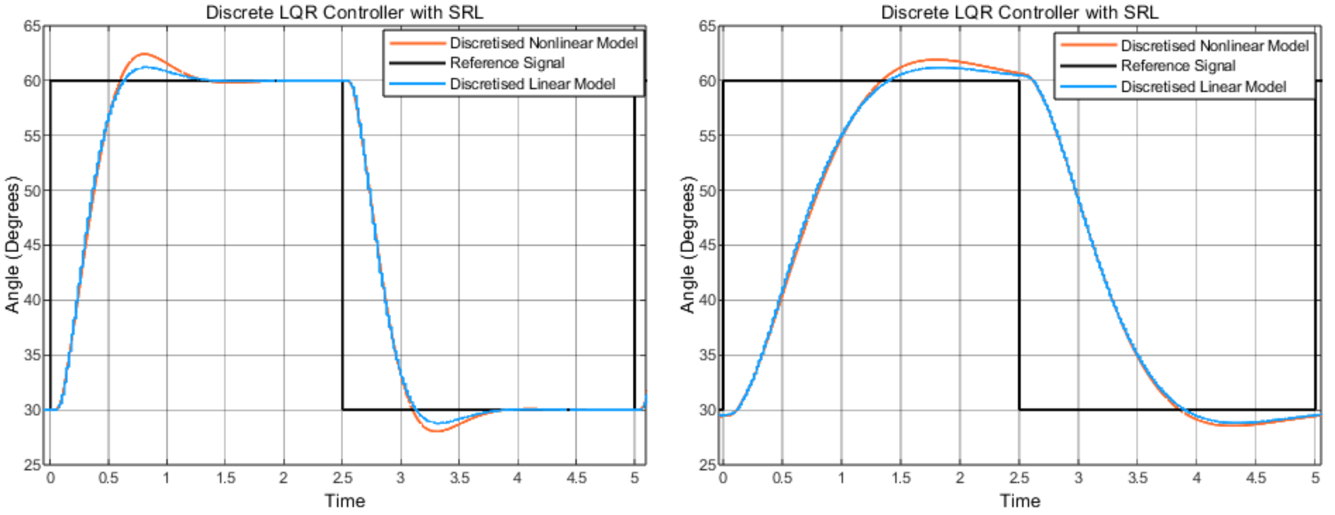
$$Q = \rho C^T C = \begin{bmatrix} 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 \end{bmatrix} \qquad R = 1 \tag{7}$$

Using these Q and R matrices, we can solve the following infinite-horizon, continuous-time Riccati equation using the LQR command in MATLAB to find the optimal gains, denoted by P in the following equation:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{8}$$

This leads to the following gains:

$$K_I = 0.4826 \qquad K = [17.9867, -12.0151, 0.3986, -0.0134, 0.0071] \tag{9}$$



(a) $\rho = 1.5$

(b) $\rho = 0.005$

Figure 3: LQR controller design using different SRL ρ values

Looking at the results of the discrete LQR controller with varying ρ values, the controller with $\rho = 1.5$ satisfies the performance requirements with a overshoot of around 3° on the non-linear plant and a settling time of 1.5s. However, when $\rho = 0.005$ the controller is unable to satisfy the settling time requirement as the poles are too slow, with the dominant pole not satisfying $\sigma \geq 2$ at this ρ value. This is also a direct result of having a R value that is much larger than the Q values, which leads to the controller weighing input costs much more than the transient performances. These differences are a direct result of non-linearity that have not been captured in the plant linearisation.

2.2 Riccati Equation Design

Another method to design the LQR controller is manually tuning the Q and R matrices such that the relative weights for each parameter is optimised in the controller performance as shown in equation 6. The R matrix represents the control effort, which is not an important design parameter in the specifications so we will set this to 1.

We will use the Q matrix calculated from the SRL as a baseline such that the settling time specification for the pole locations are satisfied. The Q matrix represents the transients of the performance output, which are key parameters in the specifications. For considerations of the weighting, the important parameters to tune is the integral state as this directly impacts the settling time and steady state error and the derivative states to reduce the overshoot response by controlling the rate of change, but leads to a slower response with a longer settling time. The proportional and current states do not impact the performance of the controller so these states are remain unchanged at 1.5. Since we can consider these states to be independent, we can solely tune the diagonal entries of the Q matrix to correspond to each state of the system.

Note that we have chosen all the terms of Q and R to be positive-definite as negative values will lead to a global maxima with a downward-concave quadratic when solving for the Riccati equation as shown in equation 8, which will be impossible to find a global minimum for. Zero values are also not included since these will lead to a family of solutions instead of a unique one when solving the Riccati equation. Following this, we tune the Q and R matrices accordingly, where the integral state is set to 100 and the derivative states are set to 200 to reduce overshoot.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 \end{bmatrix} \qquad R = 1 \tag{10}$$

Solving the Riccati equation shown in equation 8 leads to the following gains:

$$K_I = 0.4047 \qquad K = [20.8828, -15.4491, 0.4720, -0.0130, 0.0084] \tag{11}$$

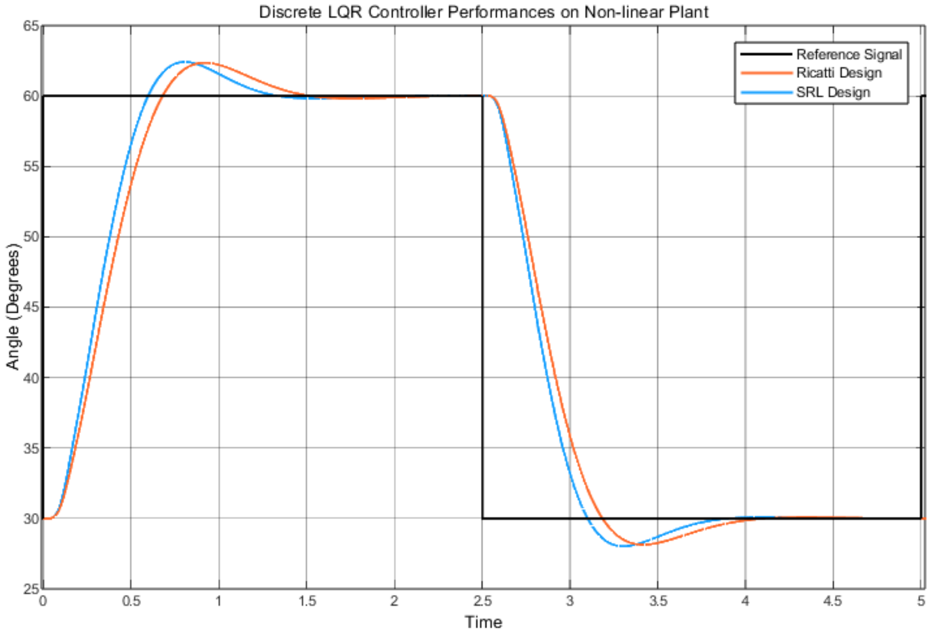


Figure 4: LQR Controller comparison between Riccati and SRL

The performances of the discrete controllers designed using SRL and Riccati methods on the non-linear plant are highly similar, with the Riccati design having a slightly longer settling time and slightly smaller overshoot due to the increased weighting on the derivative components. Overall, both methods comfortably satisfy the performance requirements of the system.

2.3 Sinusoidal Reference Tracking

Next we consider a sinusoidal signal with 30 degrees amplitude, 30 degrees offset and a period of 4s. In order to track this sinusoidal signal, we need augment the state space model to include the sinusoidal dynamics to satisfy the internal model principle. For sinusoidal signals, the model must contain the frequency of the signal, which will allow the model to account for sinusoidal signals of that frequency of varying amplitudes and phase. To do this, we define the tracking frequency as $\omega = \frac{2\pi}{4}$ and produce the model as:

$$\begin{aligned} A_d &= \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} \\ C_d &= \begin{bmatrix} 1 & 0 \end{bmatrix} \end{aligned} \tag{12}$$

Following this, we produce the augmented model of the state space with:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}'(t) \end{bmatrix} = \begin{bmatrix} A_{aug} & 0 \\ C_d^T C & A_d^T \end{bmatrix} \begin{bmatrix} x(t) \\ x'(t) \end{bmatrix} + \begin{bmatrix} B_{aug} \\ 0 \end{bmatrix} u(t) \tag{13}$$

Note that the state space representation of $\dot{x}(t)$ also contains the integral state since an offset of 30 degrees is present, which will require the internal model of the system to have an integral state to converge to this offset. After discretising this system with zero-order hold, the control input can be described as:

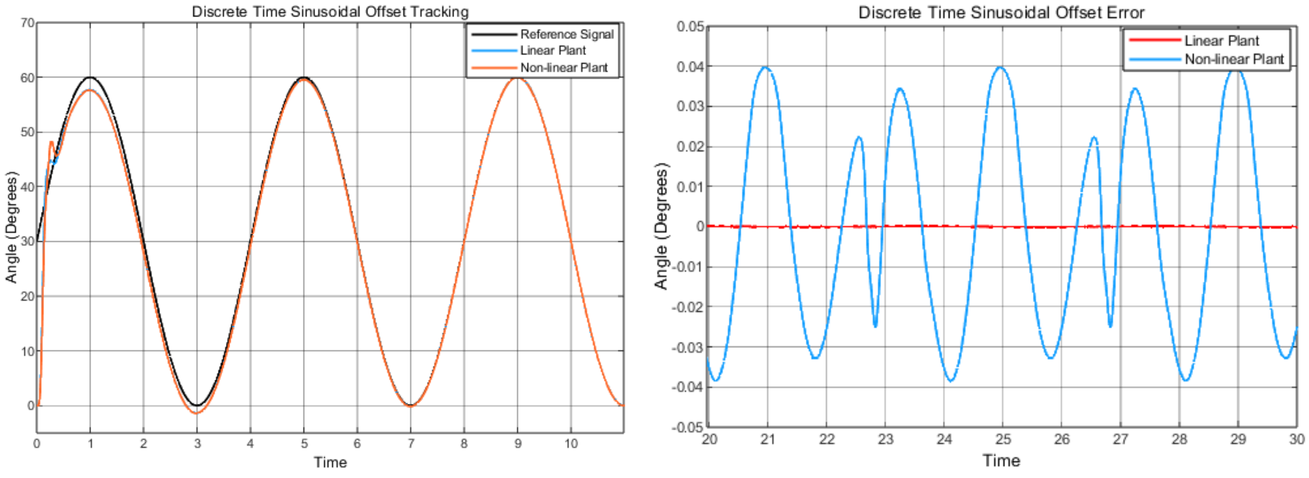
$$u(k) = -\begin{bmatrix} K_I & \mathbf{K} & K_d \end{bmatrix} \begin{bmatrix} x_I(k) \\ \mathbf{x}(k) \\ x_d(k) \end{bmatrix} \tag{14}$$

This new system will have additional states that captures the sinusoidal dynamics, therefore, we have to tune the Q and R matrices for this application. For this system, the only specification for this reference signal is to achieve zero steady-state error. From this, we can determine that the most import states to weigh for steady state performance is the integral and sinusoidal states. Note that the derivative states are also not weighted as this will reduce the rate of change of the transients, which directly impacts the performance for sinusoidal tracking. Following this, we tune the Q and R matrices accordingly, where the integral state is set to 100 as the controller does not have issue tracking the 30 degree offset, however, struggles tracking the sinusoidal signal so the sinusoidal states are set to 1000.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \qquad R = 1 \tag{15}$$

Solving the Riccati equation shown in equation 8 leads to the following gains:

$$K_I = 2.3433 \qquad K = [53.2917, 8.2250, 0.6568, 0.9684, 0.0116] \qquad K_d = [12.8931, 12.7767] \tag{16}$$



(a) Controller tracking performance

(b) Steady state error

Figure 5: LQR controller performance on sinusoidal signal with offset

The sinusoidal reference tracking performance is able to track the sinusoidal signal well, reaching steady state convergence in 10 seconds with a steady state error of ±0.04° error on the non-linear plant and negligible error on the linear plant.

3 Linear Quadratic Gaussian Control

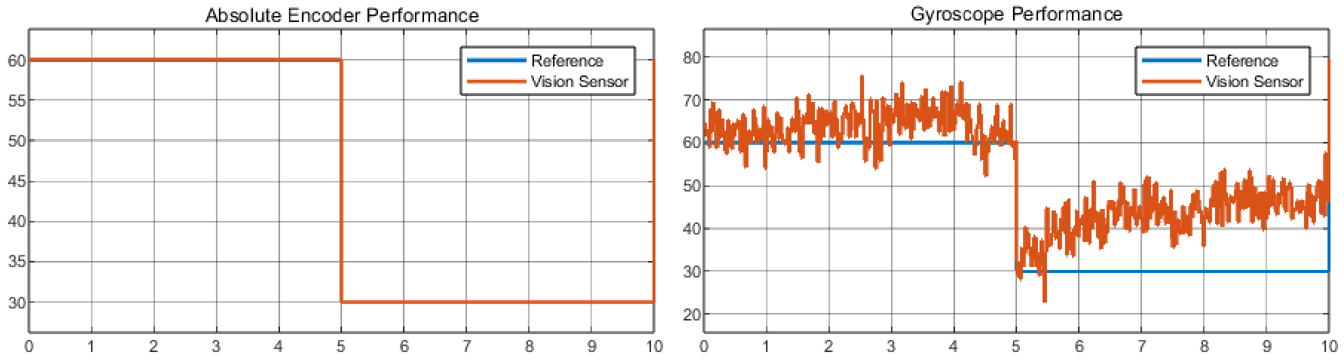
3.1 Sensor Selection

For the sensor selection, we are give a maximum budget of \$150 and a requirement to use at least 2 sensors to implement a LQG controller. The sensors of choice chosen is the CUI Devices absolute encoder with 4096 counts per revolution at \$70 and a the Invensense gyroscope at \$30 for a total cost of \$100. This results in the following C matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{17}$$

Using this C matrix, we conclude that the system is completely observable with a full column rank of 5 for each individual sensor by taking the column rank of the following observability matrix, which means a full-state observer can be developed.

$$\Gamma_c[\mathbf{A},\mathbf{C}] \triangleq \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \mathbf{CA}^3 \\ \mathbf{CA}^4 \end{bmatrix}$$



(a) Absolute encoder performance

(b) Gyroscope performance

Figure 6: Sensor performance

We can notice that the absolute encoder provides practically perfect measurements of the pulse signal, however, the gyroscope has very noisy measurements due to it's internal additive zero-mean Gaussian white-noise. Note that since this noise is zero-mean, Gaussian and independent, this satisfies the Kalman filter requirement to provide optimal full-state estimations.

3.2 Kalman Filter

Due to the fixed point linearisation approach of using $x_{2e} = 45°$ to produce a linear model instead of an Extended Kalman approach where the linearisation points dynamically change, there is substantial modelling errors in the system resulting from these assumptions. This means that the modelling noise covariance matrix W can be at 1000. For the measurement noise covariance matrix V, we consider the sensor performances in figure 6 to trust the absolute encoder more than the gyroscope, using 0.1 for the absolute encoder and 10 for the gyroscope. With this the following covariance matrices are developed:

$$W = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \qquad V = \begin{bmatrix} 0.1 & 0 \\ 0 & 10 \end{bmatrix} \tag{18}$$

We can use the duality principle and solve the Ricatti equation shown in 8 using the transposed discrete A and C matrices to find the optimal observer gains:

$$L = \begin{bmatrix} 0.9013 & 0.2878 & -7.5705 & 26.6528 & 1.5625 \\ -0.0049 & 0.0000 & -0.1646 & 1.3419 & 0.0342 \end{bmatrix}^T \tag{19}$$

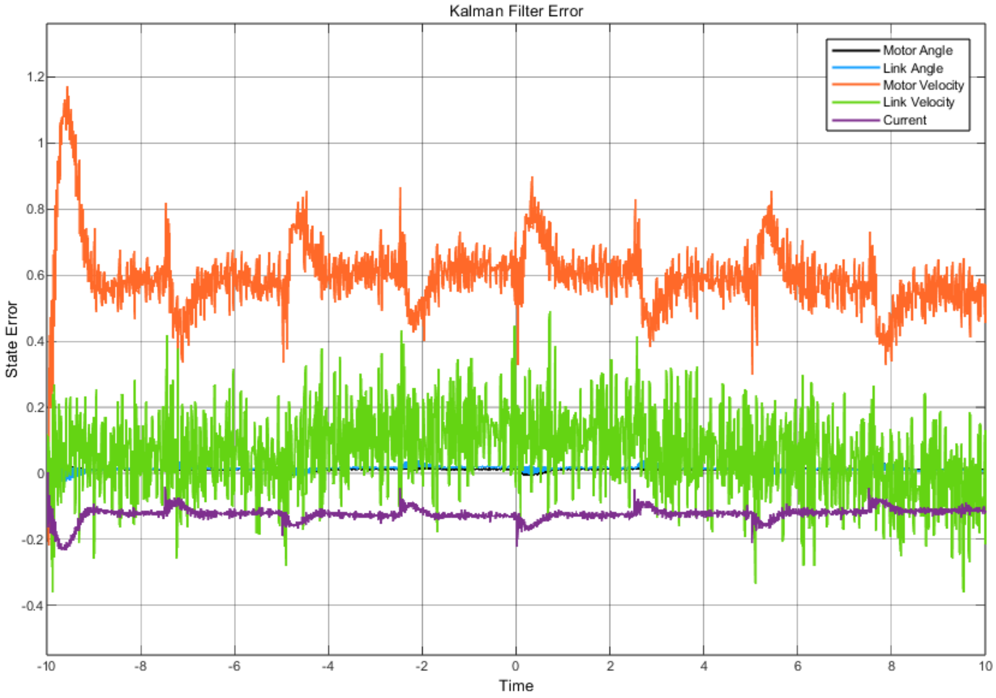


Figure 7: Kalman Filter Performance

Comparing the actual output of the non-linear plant with the Kalman filter state estimations, we see that the kalman filter is able to estimate the motor and link angles very accurately with an error of ±0.04°. The link velocity measured with the gyroscope has noisy measurements, which is expected due to the noisy measurements from the Gaussian white noise in figure 6. However, the motor velocity and current has a non-zero bias error which could be due to the non-linearities in the model resulting in non-optimal estimations of these states.

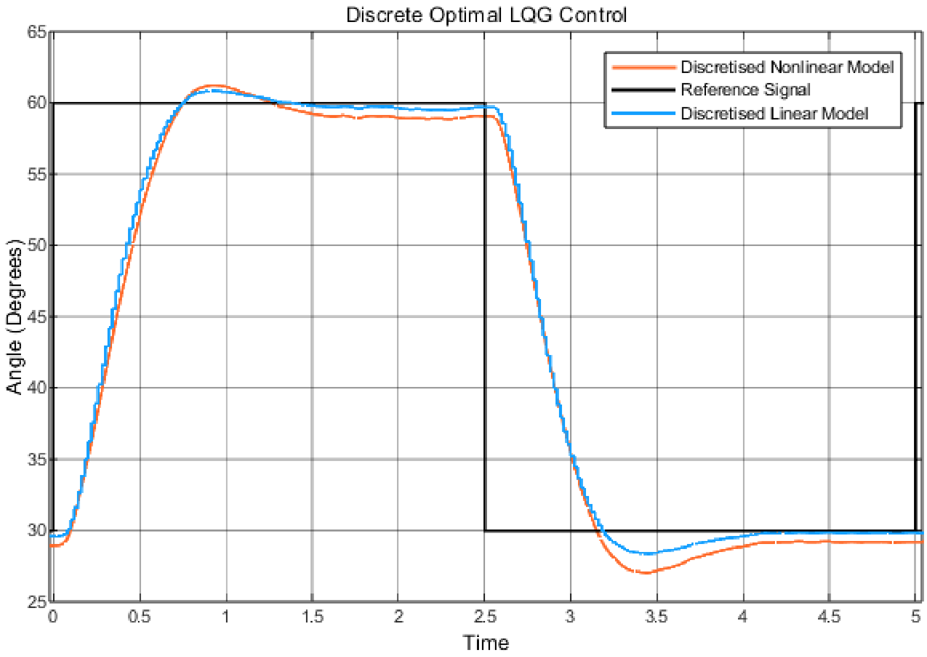


Figure 8: LQG Controls Performance

Combining this Kalman filter observer with the LQR controller developed in 11, the performance of the LQG control satisfies the specifications, with a maximum overshoot of 3°, settling time of around 1.5s and a steady state error of 1° for the non-linear model.

4 Model Predictive Control

To implement the model predictive control, we add constraints to the θ_l between 30° and 60° to have zero overshoot performance. Note that the input motor voltage is also constrained to $\pm 6V$ following the motor specifications. With this, we solve for the following optimisation problem:

$$\min_{u_k, x_k, y_k} \|W^{y_N}(y_N - r_y(t))\|_2^2 + \sum_{k=0}^{N-1} \left(\|W^y(y_k - r_y(t))\|_2^2 + \|W^u(u_k - r_u(t))\|_2^2 + \left\|W^{\Delta u}\Delta u_k\right\|_2^2 \right) \tag{20}$$

Additionally, we implement the Kalman filter observer designed in section 3.2 to feedback the full-state information into the MPC controller. A horizon length of 20 is used for the model to predict 20 iterations of the model.

To tune this optimisation problem, we will adjust the weighting matrices for penalising each state, the input changes and the terminal state. For the state penalisation a weight of 10 is placed on the θ_l and $\dot{\theta}_l$ variable as they directly affect the measured link angle, and a weight of 1 is placed on the other states.

Additionally, a weight of 0.05 is placed on penalising input changes since we are tracking a pulse signal from 30° to 60° which will require frequent input signal changes to track. Additionally, a low weight for input changes is required to remove oscillatory behaviours in the response, especially when tracking the 30° signal. This can be seen by the large fluctuations of voltage in figure 9 required to maintain the link angle at 30°. This value of 0.05 was empirically found to reduce the oscillations in the system as a value too high meant the system could not react fast enough to reduce the oscillations when tracking 30°, and a value too low meant that the system reacted too fast introducing more oscillations during tracking.

Finally, the terminal costs is set at 1 as it is not important to our system, since it ensures recursive feasibility and stability of the closed-loop system, which is not required as the system is already stable.

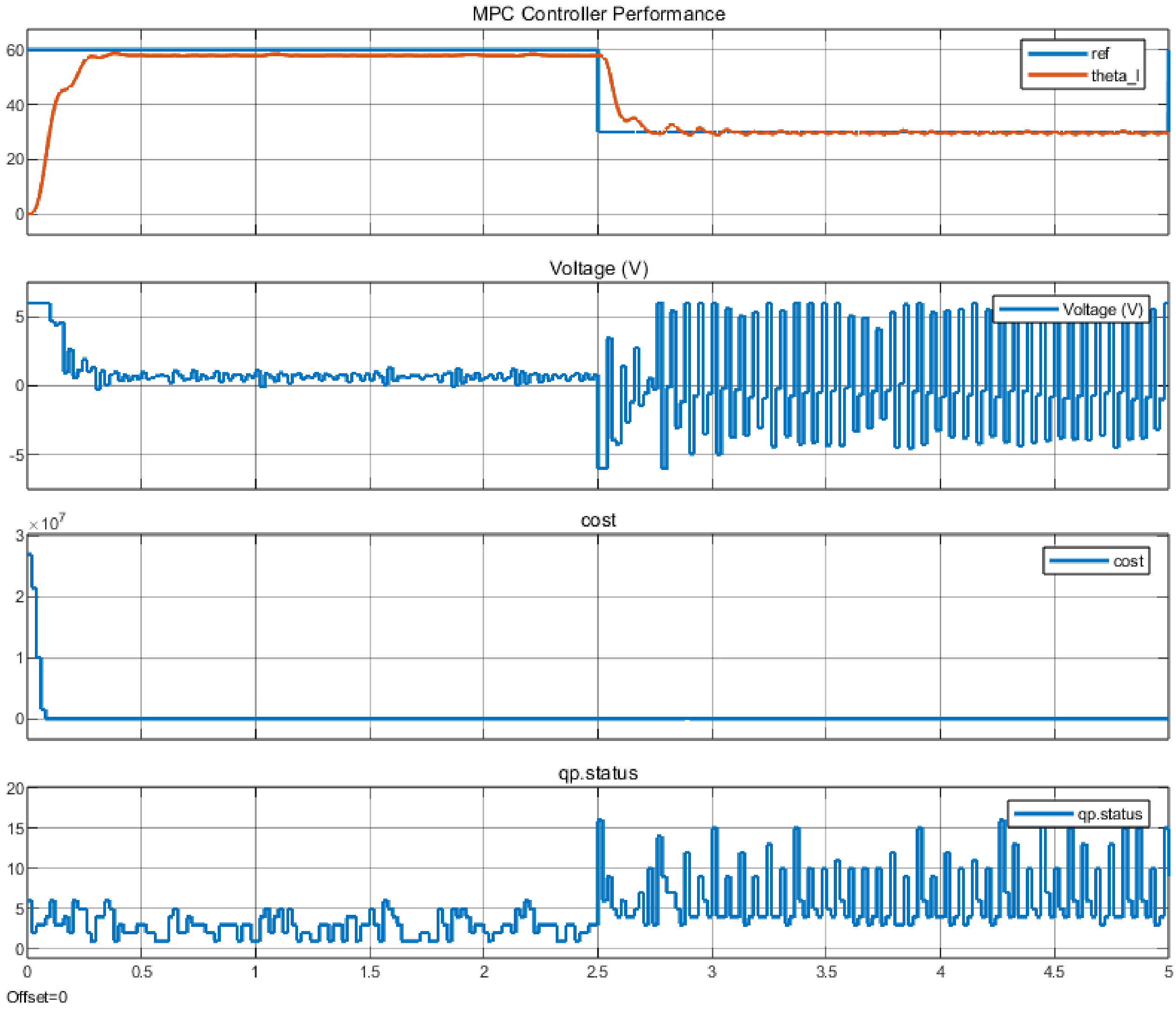


Figure 9: MPC Controller with Kalman Filter

Overall the designed MPC controller combined with the Kalman filter performs very well and does not exhibit any overshoot properties. The MPC controller is able to acheive steady state performance in around 0.5 seconds and has a steady state error of 1.8°, satisfying all the required performance requirements.