Linear Quadratic Control Design

The Linear Quadratic Regulator (LQR) and the Linear Quadratic Gaussian (LQG) control design are easy to use methods for designing controls to stabilize and regulate systems. The LQR is simply state-feedback. The LQG is used when the plant states are not directly available for measurement. It consists of two steps: the design of an LQR state-feedback controller, and the design of a Kalman-Filter observer in order to estimate the state vector. The state-feedback and the estimator are combined together to create an output feedback dynamic controller in state-space form.

1. Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) is used to design state-feedback control gains that stabilize a plant model, achieve good closed-loop performance of the states in response to transients and robustness to parameter uncertainties. It requires a plant model in state-space form. The plant inputs are the controls and the outputs are either measurements or criteria to be optimized. The plant must be stabilizable from the controls and detectable from the outputs. The control solution is a feedback from the plant states derived by the optimization of a linear quadratic performance index using the Riccati equation. The optimization takes into consideration two important and most frequently conflicting requirements: the speed of convergence of the state-vector from some initial value and the amount of the control input along a trajectory. We will present the analytic solutions for both the continuous and discrete LQR problems.

1.1 The Continuous Asymptotic LQR Problem

Equation 1.1.1 represents the plant dynamics in state-space form

$$\begin{aligned} \dot{\underline{x}}(t) &= A\,\underline{x}(t) + B\,\underline{u}(t) \\ \underline{y}(t) &= C\,\underline{x}(t) \end{aligned} \qquad (1.1.1)$$

Where:

$\underline{x}(t)$ is the state of dimension n

$\underline{u}(t)$ is the control of dimension m

$\underline{y}(t)$ is the output of dimension r

The LQR method calculates a state-feedback optimal control $u^o(t)$ that minimizes the quadratic performance index J in equation 1.1.3.

$$\begin{aligned} \underline{u}^o(k) &= -K_c\,\underline{x}(k) \\ J &= \int_0^\infty \left[y(t)^T Q\, y(t) + u(t)^T R\, u(t) \right] dt \end{aligned} \qquad (1.1.2,\ 3)$$

Where: the matrices Q and R, are defined to be the output and control weight matrices. They are used as knobs which adjust the closed-loop system's response to initial states $\underline{x}_0$ or to disturbances. They trade performance in the output vector y(t) in terms of speed of convergence versus the magnitudes of control u(t). The (r x r) matrix Q should be symmetric, positive semidefinite, and the matrix R (m x m) should be symmetric positive definite.

Selecting a small R or a large Q in the LQR design, it is telling the mathematics that when the control loop is closed and the system is initialized at some initial state x(0), I would like the output response $\underline{y}$(t) to convergence fast to the commanded value with small transients, regardless of how much control force $\underline{u}$(t) is necessary to achieve this. This results in a high bandwidth control system and possibly effector saturation.

If on the other hand a large R or small Q are used in equation 1.1.3, the magnitudes of the control $\underline{u}$(t) are penalized more than the output transients $\underline{y}$(t) in the performance index. It indicates that my actuators do not have as much strength to handle disturbances or big commands. The closed-loop system's response to disturbances will be slower, resulting in a reduced control system bandwidth that will protect the actuators from saturating.

The solution to the above problem exists if the (A, B) system is stabilizable, and the (A, D) pair is detectable where D is defined by equation 1.1.4. it means that all unstable plant modes must be controllable and measurable.

$$D^T D = C^T Q C \qquad (1.1.4)$$

Notice, that the variable y(t) used in the optimization criterion is not necessarily the plant output. Any combination of output variables, not necessarily measurable, represented by a matrix C_1, different than C, can be used in the optimization criterion, as long as the plant states are detectable from C_1. C_1 may also be the identity matrix, in which case the state variables are directly and individually penalized in the performance index via matrix Q.

The state-feedback gain K_c of equation 1.1.2 is calculated from equation 1.1.5, and the (n x n) matrix P is obtained from the steady-state solution of the Riccati Equation 1.1.6

$$\begin{aligned} K_c &= R^{-1} B^T P \quad where: \\ -\dot{P} &= P A + A^T P + C^T Q C - P B R^{-1} B^T P = 0 \end{aligned} \qquad 1.1.5,\ 6$$

Furthermore if the problem is initialized with an initial state error $\underline{x}(0)=\underline{x}_0$, then the performance index criterion is: $J = x(0)^T P\, x(0)$

1.2 The Discrete Asymptotic Case

In the discrete case the plant system is represented by the difference matrix equations 1.2.1.

$$x(k+1) = A\,x(k) + B\,u(k)$$
$$y(k) = C\,x(k) \qquad (1.2.1)$$

Where:

$\underline{x}(k)$ is the state of dimension n

$\underline{u}(k)$ is the control of dimension m

$\underline{y}(k)$ is the output of dimension r

k represents the present state variable and (k+1) is the next iteration

The Discrete LQR method calculates a state-feedback optimal control $u^o(k)$ that minimizes the quadratic performance index J in equation 1.2.2.

$$\underline{u}^o(k) = -K_c\,\underline{x}(k)$$
$$J = \lim(N \to \infty)\sum_{0}^{N-1}\left[y(k+1)^T Q\,y(k+1) + u(k)^T R\,u(k)\right] \qquad (1.2.2\text{- }3)$$

Where: the matrices Q and R, are defined as in the continuous case. The (m x n) state feedback gain matrix Kc is obtained from equation 1.2.4, where: the (n x n) matrix P is obtained from the solution of the discrete steady-state Riccati Equation 1.2.5

$$K_c = \left(R + B^T P_{k+1}\,B\right)^{-1} B^T P_{k+1}\,A$$
$$P_k = \left[A - B\,K_c\right]^{-1} P_{k+1}\left[A - B\,K_c\right] + K_c^T R\,K_c + C^T Q\,C = 0 \qquad (1.2.4\text{- }5)$$

Where: k= 0, 1, 2, 3, …, N-1

The solution to the above problem exists if the (A, B) system is stabilizable, and the (A, D) pair is detectable, where D is defined by equation 1.2.6. This means that all unstable plant modes must be controllable and measurable.

$$D^T D = C^T Q\,C \qquad (1.2.6)$$

Furthermore if the problem is initialized with an initial state error $\underline{x}(0)=\underline{x}_0$, then the performance index criterion is: $J = x(0)^T P(0)\,x(0)$

2. The Finite-Time LQR with Terminal State Penalty

The finite-time or transient deterministic optimal linear quadratic regulator problem is essentially similar to the steady-state case described in section 1. The difference is that an additional term is included in the performance index that penalizes the value of the state-vector $x(t_f)$ at the terminal time t_f. The resulting state-feedback control law Kc(t) is time-varying.

2.1 The Continuous Transient LQR Problem

The plant dynamics in state-space form is the same as before and the system is initialized at $\underline{x}(0)=\underline{x}_0$

$$\begin{aligned} \dot{\underline{x}}(t) &= A\,\underline{x}(t) + B\,\underline{u}(t) + w(t) \\ \underline{y}(t) &= C\,\underline{x}(t) \end{aligned} \qquad (2.1.1)$$

Where:

$\underline{x}(t)$ is the state of dimension n

$\underline{u}(t)$ is the control of dimension m

$\underline{y}(t)$ is the output of dimension r

w(t) is white noise with intensity V(t)

The Transient LQR solution calculates a state-feedback optimal control $u^o(t)$ that minimizes the quadratic performance index J in equation 2.1.3.

$$\underline{u}^0(t) = -K_c(t)\,\underline{x}(t) \quad where:$$

$$J = \int_0^{t_f} \left[\underline{y}(t)^T Q\, \underline{y}(t) + \underline{u}(t)^T R\, \underline{u}(t) \right] dt + \underline{x}(t_f)^T P_1\, \underline{x}(t_f) \qquad (2.1.2\text{- }3)$$

Where: (t_f) is a known terminal time. The matrices Q and R are the output and control weight matrices as already described in Section 1. P_1 is a weight matrix that penalizes the terminal state. These matrices determine the optimal trade-off between: the output y(t) deviations from zero along the trajectory, the magnitude of control input u(t), and the dispersion of the terminal state vector $\underline{x}(t_f)$ from zero at the end-time.

Matrix Q is (r x r) and should be symmetric positive semidefinite,

Matrix R is (m x m) and should be symmetric positive definite,

Matrix P_1 is (n x n) and should be symmetric positive semidefinite

The (m x n) state feedback gain matrix $K_c(t)$ is obtained from equation 2.1.4, where: the (n x n) matrix P(t) is obtained from the solution of the transient Riccati Equation 2.1.5

$$K_c(t) = R^{-1} B^T P(t) \ \ where: \qquad (2.1.4, 5)$$
$$\dot{P} = P A + A^T P + C^T Q C - P B R^{-1} B^T P$$

The equation 2.1.5 is solved backwards in time after being initialized at the terminal time t_f where: $P(t_f) = P_1$

Furthermore the performance index criterion J is

$$J = \underline{x}(0)^T \ P \, \underline{x}(0) + \int_0^{t_f} trace\big[P(t) V(t)\big] dt \qquad (2.1.6)$$

Solution of the Continuous Transient LQR

The following algorithm, from reference [1], is used to solve the transient regulator problem. Let us define a matrix Z where:

$$Z = \begin{bmatrix} A & -B R^{-1} B^T \\ -C^T Q C & -A \end{bmatrix} \qquad (2.1.7)$$

Matrix Z has the property that if α is an eigenvalue of Z, $-\alpha$ is also an eigenvalue of Z. We can find the eigenvector matrix W such that

$$Z = W \begin{bmatrix} \Lambda & 0 \\ 0 & -\Lambda \end{bmatrix} W^{-1} \qquad (2.1.8)$$

Where: Λ is a diagonal matrix consisting of the positive eigenvalues of Z, and $-\Lambda$ consisting of the negative eigenvalues of Z. Then we partition the (2n x 2n) matrix W into four (n x n) blocks as shown in equation 2.1.9

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} \qquad (2.1.9)$$

The solution P(t) of the Riccati equation 2.1.5 is obtained from equation 2.1.10, where

$$P(t) = \big[W_{22} + W_{21}\, G(t_f - t)\big]\big[W_{12} + W_{11}\, G(t_f - t)\big]^{-1} \ where:$$
$$G(t) = \exp(\Lambda t)\, S \exp(-\Lambda t), \ and \qquad (2.1.10\text{-}12)$$
$$S = -\left(W_{22} - P_1 W_{12}\right)^{-1} \left(W_{21} - P_1 W_{11}\right)$$

From equation (2.1.11) as (t_f) approaches infinity, $G(t_f\text{-}t)$ approaches zero, and the steady-state solution of the Riccati equation becomes: $P = W_{22}\, W_{12}^{-1}$

2.2 The Discrete Time Transient LQR

In the Discrete Transient LQR design case the plant system is represented by the difference matrix equations 2.2.1

$$x(k+1) = A\,x(k) + B\,u(k) + w(k)$$
$$y(k) = C\,x(k) \qquad (2.2.1)$$

Where:

$\underline{x}(k)$ is the state of dimension n
$\underline{u}(k)$ is the control of dimension m
$\underline{y}(k)$ is the output of dimension r
w(k) is zero mean white noise with variance V(t)

We must calculate a state-feedback optimal control $u^o(k)$ that minimizes the quadratic performance index J in equation 2.2.3.

$$\underline{u}^o(k) = -K_c(k)\,\underline{x}(k)$$
$$J = \lim(N \to \infty)\sum_{0}^{N-1}\left[\underline{y}(k+1)^T Q\,\underline{y}(k+1) + \underline{u}(k)^T R\,\underline{u}(k)\right] + \underline{x}(N)^T\, P_N\, \underline{x}(N) \qquad (2.2.2,\ 3)$$

The matrices Q and R and P_N are defined as in the continuous case. The discrete optimal regulator solution is obtained from the difference equation 2.2.4 solved backwards in time, initialized at N with $P(N)=P_N$. The (n x n) matrix P(k) is obtained from the solution of the discrete transient Riccati Equation 2.2.5, where: k=0, 1, 2,…, N-1

$$K_c(k) = \left(R + B^T P(k+1)\,B\right)^{-1} B^T P(k+1)\,A$$
$$P(k) = \left[A - B\,K_c(k)\right]^{-1} P(k+1)\left[A - B\,K_c(k)\right] + K_c^T(k) R\,K_c(k) + C^T Q\,C = 0 \qquad (2.2.4,\ 5)$$

Furthermore if the problem is initialized with an initial state error $x(0)= x_0$, then the performance index criterion is

$$J = \underline{x}(0)^T\ P(0)\,\underline{x}(0) + \sum_{j=0}^{j=N-1} trace\left[V(j)P(j+1)\right] \qquad (2.2.6)$$

3. The Asymptotic Kalman-Bucy State-Estimator, Observer

In the previous two sections we demonstrated how to design optimal state-feedback controllers, assuming that the state vector can be measured accurately and be available for feedback. This assumption is most often unrealistic because in most systems the state vector is not directly measurable but the output measurements consist of a linear combination of the states. We therefore need to design an observer, which is a system that will approximately reconstruct the state vector from the plant output, and this will allow us to apply our optimal state-feedback control laws. In this section we will present the design of the steady-state Kalman-Bucy filter that is used to reconstruct an approximation of the state vector from the measured system output that will converge to the state vector. We shall also assume that the system is corrupted by two types of noises: measurement, and state excitation noise. They are both white, zero mean and are not correlated. We will first analyze the continuous and then the discrete Kalman-Filter observer for continuous and discrete systems.

3.1 The Continuous Kalman-Bucy Filter

Let us consider the following plant in state-space form. This system is affected by disturbances $\underline{w}$(t) and the observations $\underline{y}$(t) are corrupted by noise $\underline{v}$(t)

$$\begin{aligned} \dot{\underline{x}}(t) &= A\,\underline{x}(t) + B\,\underline{u}(t) + G\,w(t) \\ \underline{y}(t) &= C\,\underline{x}(t) + v(t) \end{aligned} \qquad (3.1.1)$$

Where:

$\underline{x}$(t) is the state vector of dimension n.
$\underline{u}$(t) is the control input vector of dimension m.
w(t) is the process noise of dimension l ($l \le n$) and covariance matrix Q_{pn}, where $Q_{pn}=Q'_{pn}\ge 0$.
$\underline{y}$(t) is the measurement vector of dimension r ($r \le n$).
$\underline{v}$(t) is the measurement noise with intensity $R_{mn}=R'_{mn}\ge 0$.

The purpose of the optimal Kalman Filter estimator is to construct an estimate of the state x operating over the time range [t_0->t] such that the index J in equation 3.1.2 is minimized, where: $\hat{x}(t)$ denotes the estimate of x(t). E is the expected value, and the matrix W is (n x n) positive semi-definite.

$$J = \lim(t_0 \to \infty)\; E\left[\left(\underline{x}(t) - \hat{\underline{x}}(t)\right)^T W \left(\underline{x}(t) - \hat{\underline{x}}(t)\right)\right] \qquad (3.1.2)$$

The solution exists when the pair (A^T,C^T) is stabilizable and the pair (A,D) is detectable
Where: $D\,D' = G\,Q_{pn}\,G^T$

After initializing with the expected value of the initial state: $\underline{\hat{x}}(0) = E[\underline{x}(0)]$, the state estimate is computed by the following differential equation 3.1.4. This equation can also be written as in 3.1.5 to show that the inputs to the estimator are the plant control vector u(t) and the measurements y(t), as shown in figure (3.1)

$$\dot{\underline{\hat{x}}}(t) = A\underline{\hat{x}}(t) + B\underline{u}(t) + K_f\left[\underline{y}(t) - C\underline{\hat{x}}(t)\right]$$
$$\dot{\underline{\hat{x}}}(t) = \left[A - K_f C\right]\underline{\hat{x}}(t) + B\underline{u}(t) + K_f\,\underline{y}(t) \qquad (3.1.4, 5)$$

The steady-state Kalman-Filter gain K_f is obtained from equation 3.1.6, where matrix P is symmetric positive semi-definite and is obtained from the steady-state solution of the Asymptotic Riccati Equation 3.1.7

$$K_f = P C^T R_{mn}^{-1} \;\; where:$$
$$\dot{P} = A P + P A^T + G Q_{pn} G^T - P C^T R_{mn}^{-1} C P = 0 \qquad (3.1.6, 7)$$

The mean square reconstruction error is shown in equation 3.1.8

$$J = \lim(t \to \infty)\;\; E\left[\left(\underline{x}(t) - \underline{\hat{x}}(t)\right)^T W\left(\underline{x}(t) - \underline{\hat{x}}(t)\right)\right] = trace\left[P W\right] \qquad (3.1.8)$$

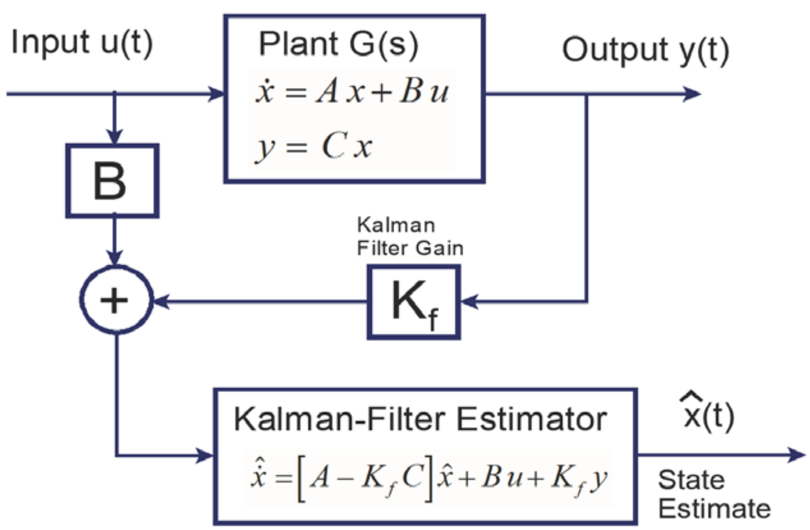


Figure 3.1 Functional Block Diagram of the Kalman-Filter Steady-State Observer

The optimal observer described provides a compromise between the speed of state reconstruction and the immunity to measurement noise. The balance between these two properties is determined by the magnitudes of the white noise intensity matrices Q_{pn} and R_{mn} that can be adjusted to satisfy design requirements. Decreasing R_{mn} and increasing Q_{pn} improves the speed of state reconstruction and shifts the observer poles further to the left side of the complex plane but the observer becomes more vulnerable to observation noise.

3.2 The Discrete-Time Kalman-Bucy Filter

The steady-state Kalman-Filter estimator for the discrete plant is obtained in a similar fashion. The dynamic model for the discrete plant is defined by the difference equations 3.2.1, where the vectors x, w, v, and y are defined as in the continuous equation 3.1.1.

$$\begin{aligned} x(k+1) &= A\,x(k) + B\,u(k) + G\,w(k) \\ y(k) &= C\,x(k) + v(k) \end{aligned} \qquad (3.2.1)$$

The discrete optimal Kalman Filter estimator problem is to construct an estimate of the state $\hat{x}(k)$ from previous measurements of the output vector [y(0), y(1)...y(k-1)] such that the quantity J in equation 3.2.2 is minimized, where E denotes the expected value, and W is (n x n) positive semi-definite matrix.

$$J = \lim(k_0 \to \infty)\ E\left[\left(\underline{x}(k) - \underline{\hat{x}}(k)\right)^T W \left(\underline{x}(k) - \underline{\hat{x}}(k)\right)\right] \qquad (3.2.2)$$

The solution exists when the pair (A^T, C^T) is stabilizable and the pair (A, D) is detectable
Where: D D'= G Q_{pn} G^T

After initializing with the expected value of the initial state: $\underline{\hat{x}}(0) = E\left[\underline{x}(0)\right]$, the state estimate is computed by the following difference equation 3.2.4. This equation can also be written as in 3.2.5 to show that the inputs to the estimator are the plant control vector $\underline{u}$(k) and the measurements $\underline{y}$(k).

$$\begin{aligned} \underline{\hat{x}}(k+1) &= A\underline{\hat{x}}(k) + B\underline{u}(k) + K_f\left[\underline{y}(k) - C\,\underline{\hat{x}}(k)\right] \\ \underline{\hat{x}}(k+1) &= \left[A - K_f C\right]\underline{\hat{x}}(k) + B\,\underline{u}(k) + K_f\,\underline{y}(k) \end{aligned} \qquad (3.2.4, 5)$$

The Kalman-Filter gain K_f is calculated from equation (3.2.6), where matrix P represents the steady-state variance of the state-vector reconstruction error. It is symmetric positive semi-definite and is obtained by solving asymptotically the recursive Riccati Equation 3.2.7.

$$\begin{aligned} K_f &= A\,P\,C^T\left(R_{mn} + C\,P\,C^T\right)^{-1} \quad where: \\ P_{k+1} &= \left(A - K_f C\right)^T P_k \left(A - K_f C\right) + G\,Q_{pn} G^T + K_f\,R\,K_f^T \end{aligned} \qquad (3.2.6, 7)$$

The initial state of the estimator must be set equal to the plant state: $\underline{\hat{x}}(0) = \underline{x}_0$. The following result is also true

$$J = \lim(k_0 \to \infty)\ E\left[\left(\underline{x}(k) - \underline{\hat{x}}(k)\right)^T W \left(\underline{x}(k) - \underline{\hat{x}}(k)\right)\right] = trace[PW]$$

4. Linear Quadratic Gaussian Output Feedback Control

The Linear Quadratic state-feedback controller and the Kalman-Filter results obtained from Sections 1 and 3 are now combined together to create an output feedback dynamic controller. Let us again consider the state-space plant model that we want to control.

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + B\underline{u}(t) \\ \underline{y}(t) &= C\underline{x}(t) \end{aligned} \qquad (4.1.1)$$

Where:

$\underline{x}(t)$ is the plant state of dimension n

$\underline{u}(t)$ is the plant control input of dimension m

$\underline{y}(t)$ is the plant output of dimension r

The optimal steady-state, state-feedback control $\underline{u}^o(t) = -K_c\underline{x}(t)$ was derived in Section 1, and the Kalman-Filter observer was described in Section 3.

$$\begin{aligned} \dot{\hat{\underline{x}}}(t) &= A\hat{\underline{x}}(t) + B\underline{u}(t) + K_f\left[\underline{y}(t) - C\hat{\underline{x}}(t)\right] \\ \dot{\hat{\underline{x}}}(t) &= \left[A - K_f C\right]\hat{\underline{x}}(t) + B\underline{u}(t) + K_f\,\underline{y}(t) \end{aligned} \qquad (4.1.3, 4)$$

Since the state vector x(t) is not directly available for measurement, we will use the estimated state vector and apply the control feedback trough the estimated state rather than the actual state

$$\underline{u}^o(t) = -K_c\hat{\underline{x}}(t) \qquad (4.1.5)$$

The block diagram in Figure 4.1a shows the state-estimator and the state-feedback controller operating in closed loop form around the plant. Figure 4.1.b is the same system but the observer and the state-feedback gain are combined together as a single dynamic control system that provides feedback from the plant output instead of the states. The closed loop dynamic model is obtained by combining the plant and Kalman-Filter equations in a (2n x 2n) system 4.1.6.

$$\begin{pmatrix} \dot{\underline{x}} \\ \dot{\hat{\underline{x}}} \end{pmatrix} = \begin{bmatrix} A & -BK_C \\ K_f C & A - K_f C - BK_C \end{bmatrix}\begin{pmatrix} \underline{x} \\ \hat{\underline{x}} \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix}\underline{u} \qquad (4.1.6)$$

After considering the state reconstruction error $\underline{e}(t) = \underline{x}(t) - \hat{\underline{x}}(t)$ we obtain equation 4.1.7

$$\begin{pmatrix} \dot{\underline{x}} \\ \dot{\underline{e}} \end{pmatrix} = \begin{bmatrix} A - BK_C & -BK_C \\ 0 & A - K_f C \end{bmatrix}\begin{pmatrix} \underline{x} \\ \underline{e} \end{pmatrix} \qquad (4.1.7)$$

The eigenvalues of the system in equation 4.1.7 consist of the eigenvalues of det(sI-A+BK_c) plus the eigenvalues of det(sI-A+K_fC). Consequently the closed loop system comprises the eigenvalues of the optimal controller under state feedback, plus the eigenvalues of the observer. This is an important principle called the “separation principle", because, if we design a stable state-feedback controller and an asymptotically stable observer independently, the resulting interconnection system, equations (4.1.6) or (4.1.7), is an asymptotically stable system.

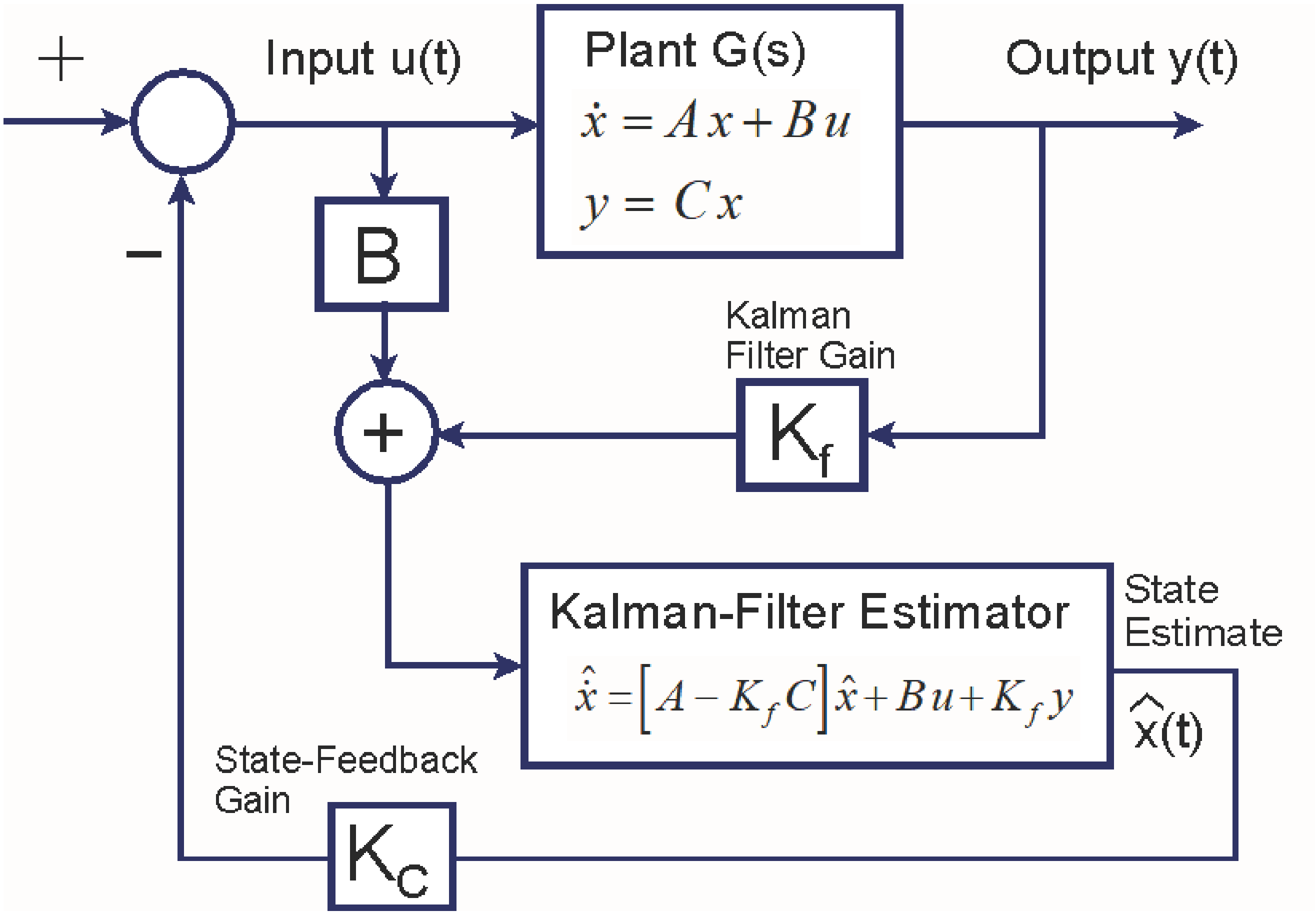


Figure 4.1a Structure of the Output Feedback Dynamic Controller Consisting of the Kalman-Filter Estimator and the State-Feedback Gain Kc

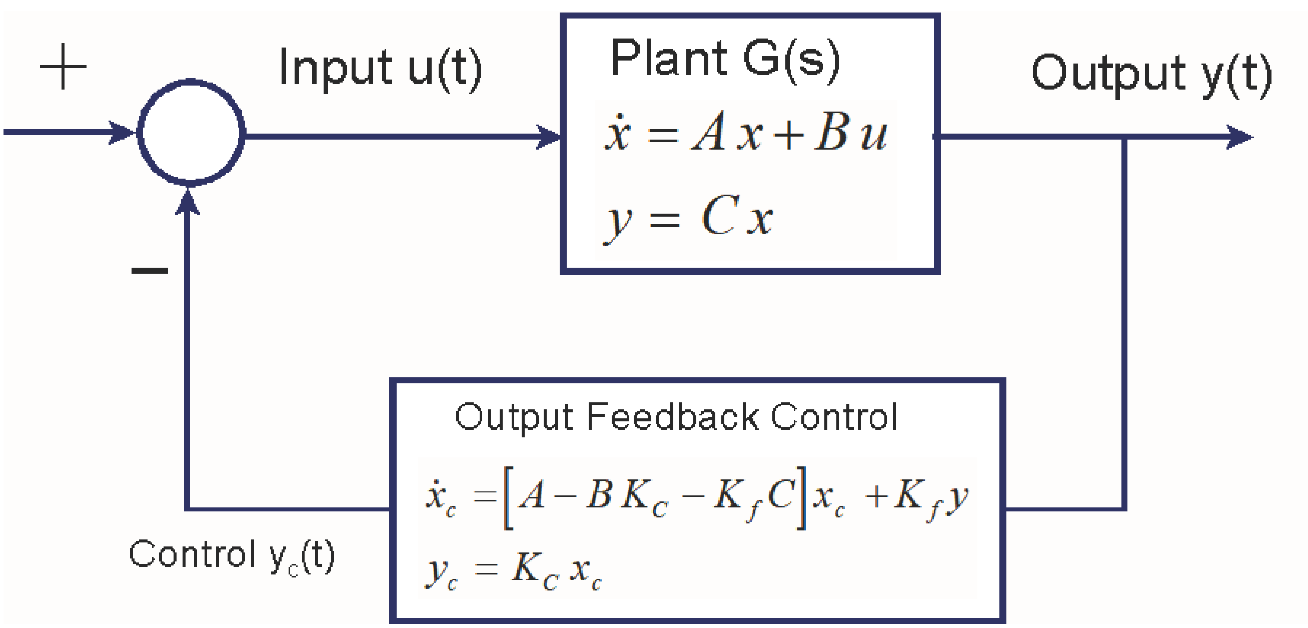


Figure 4.1b Closed-Loop System with Dynamic Controller in State-Space Form Providing Feedback from the Plant Output

5. Linear Quadratic Control Program

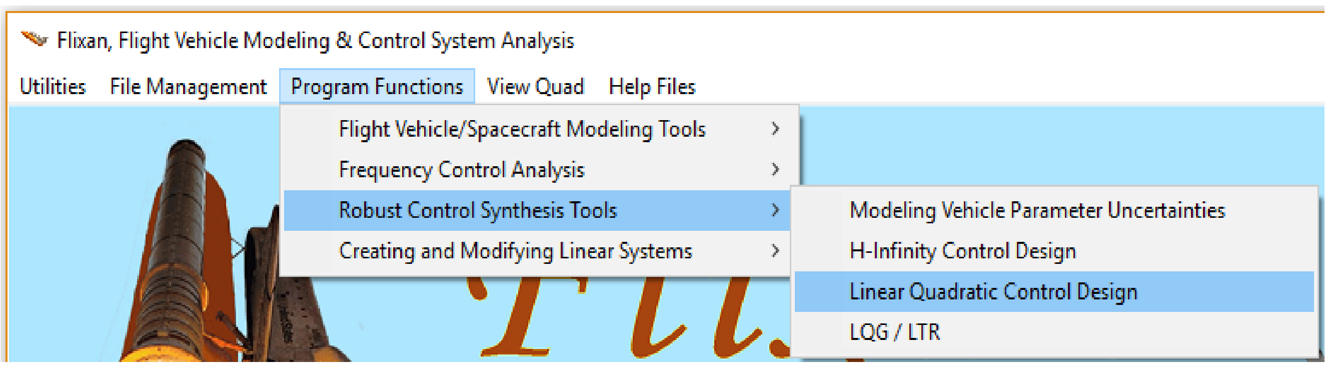
The Flixan Linear Quadratic Control program implements the four LQG functions described in Sections 1 through 4. The user must provide the plant model G(s) and the weight matrices described which must be included in a systems file (.Qdr). The matrices can also be entered interactively. The program calculates the control matrix Kc, the estimator gain Kf or the LQG dynamic control system K(s) and saves them in the same systems file. It runs either interactively or in batch mode. When in batch mode it processes input datasets from an already created input file (.Inp). The dataset of an operation is automatically created and saved in the input file after running it interactively the first time. It can be reprocessed multiple times in batch mode which is much faster than interactively. Typically the user runs initially the Flixan programs interactively to create the datasets and then when he needs to make a modification in the data he may reprocess the datasets in batch mode, either each set individually or the entire file using a batch set. The program consists of the following options:

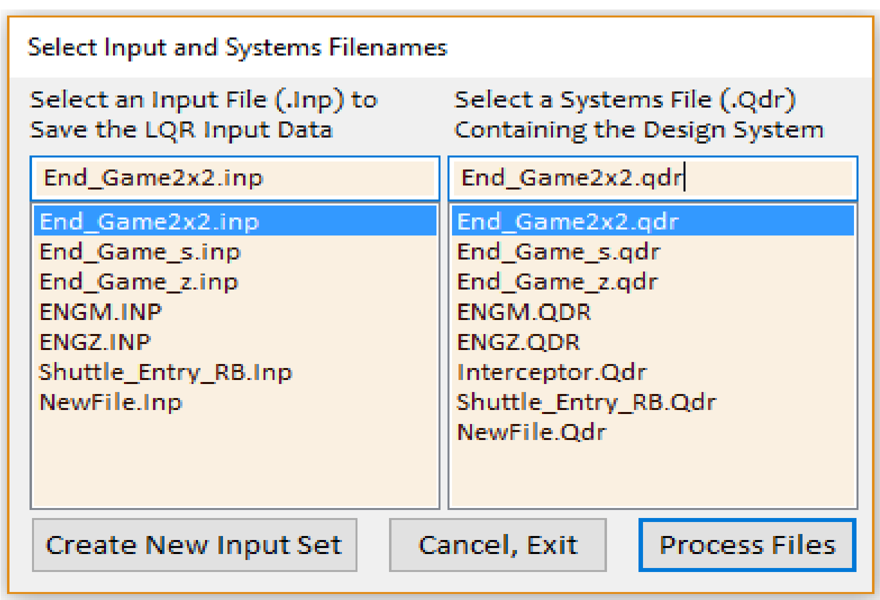
1. The Asymptotic LQR design for a continuous or discrete time plant described in Section 1. The program reads the continuous G(s) or discrete G(z) plant state-space model from the systems file, the output weighting matrix Q_c, and the control weighting matrix R_c. It solves either the continuous or the discrete LQR problem depending on the plant sampling period, which is zero when the plant is continuous. It calculates the steady-state LQR state-feedback gain Kc and saves it in the systems file (.Qdr). The user may choose between two algorithms for solving the asymptotic Riccati equation.
2. The Transient LQR design for a continuous or discrete time plant described in Section 2. The program reads the continuous G(s) or discrete G(z) plant state-space model from the systems file, the (r x r) output weighting matrix Q_c, the (m x m) control weighting matrix R_c, and the (nxn) weighting matrix P_1 that penalizes the state vector at the terminal time t_f. It requires also the initial time-to-go before the final time, and the number of points to calculate the state-feedback gains (only when the plant is continuous). It calculates the time-varying gain matrix Kc(t) and saves it as a function of time-to-go in Excel format in file “Gains.Txt”. Tgo is in the first column and the gain matrix is printed in rows.
3. The Kalman-Filter State Estimator for a continuous or discrete time plant described in Section 3. The program reads the continuous G(s) or discrete G(z) plant state-space model from the systems file, the (nxl) input noise matrix G, the (rxr) process noise intensity matrix Q_{pn}, and the (mxm) measurement noise covariance matrix R_{mn}. It solves either the continuous or the discrete KF observer problem, calculates the (nxr) Kalman-Filter gain K_f and saves it in the systems file (.Qdr).
4. The Dynamic Output Feedback Controller that is described in Section 4. The program combines the results obtained in steps 1 and 3, which are: the state-feedback gain Kc and the Kalman-Filter gain K_f, to synthesize a steady-state control system in the situation when the state-vector is not available for measurement. It reads the two matrices from the systems file (.Qdr), calculates the output-feedback controller K(s) in state-space form and saves it in the same systems file.

The in-between program calculations, such as matrix P, errors in the Riccati solution, closed-loop system eigenvalues, etc. are saved in file "LQC.Out" after execution. The analyst may check this file to make sure that no errors have occurred, eigenvalues are stable, matrix P is symmetric, etc. It is also important to check the plant's controllability and observability because the success of the solutions depends on that. It is the first option in the menu of the Linear Quadratic Control design program and it is only available in the interactive version when you begin analyzing the system, since it is not necessary to rerun it interactively when you reprocess the dataset in batch mode.

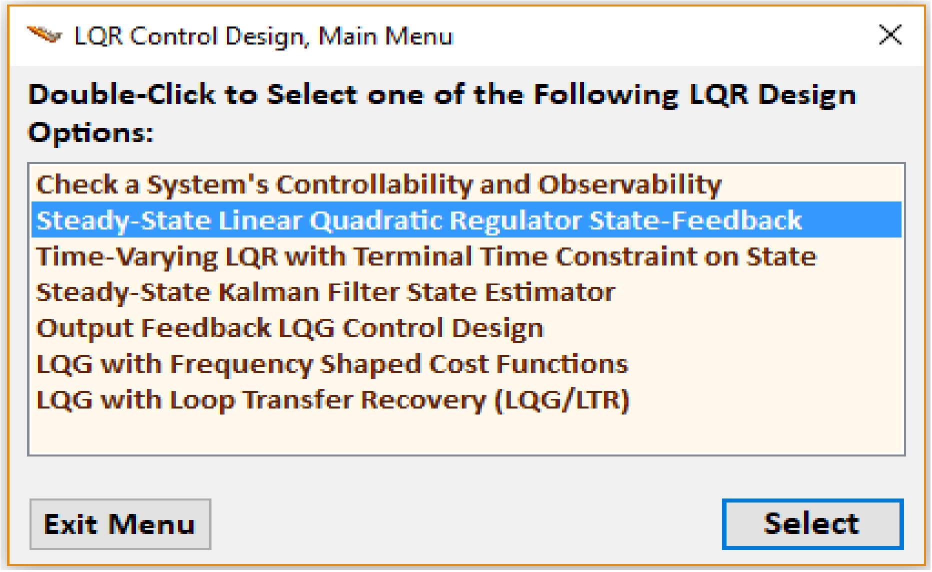
5.1 Running the Program Interactively

The Linear Quadratic Control design program is selected from the Flixan main menu by going to "*Program Functions*", "*Robust Control Synthesis Tools*", and then "*Linear Quadratic Control Design*", as shown below. Select the input filename to save the operation dataset and the systems filename where it will read and write the systems and matrices, and click on "*Process Files*".

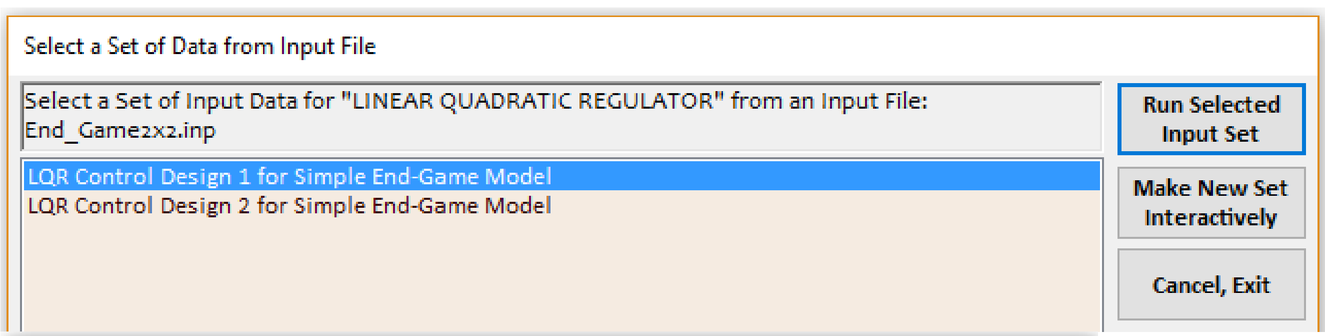


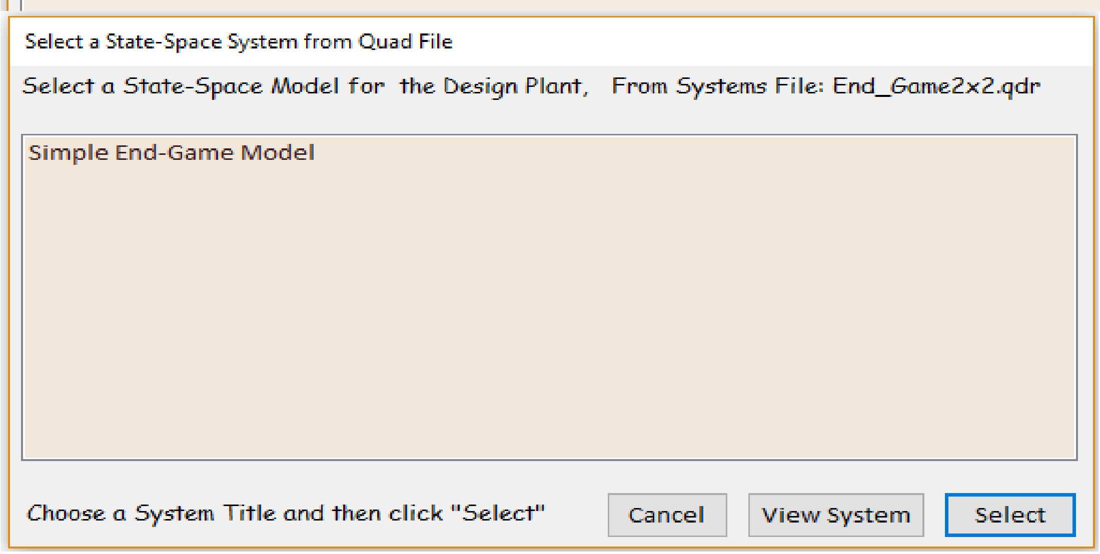


The main LQR control design menu includes several options. Select one of the options, such as: Steady-State LQR design in this case, and click on "*Select*".

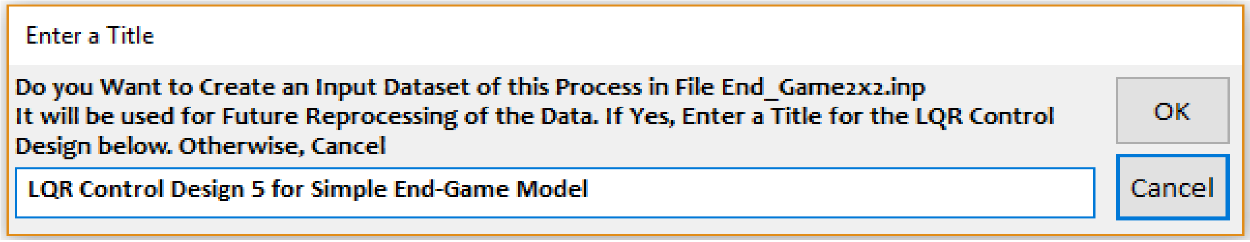


In this example the input file already contains 2 LQR design datasets. You can either process one of the 2 existing datasets or you can create a new dataset by clicking on "*Make a New Set Interactively*". Choose the second option and from the next menu select the title of the plant model that will be stabilized by LQR, "*Simple End-Game Model*" in this example.

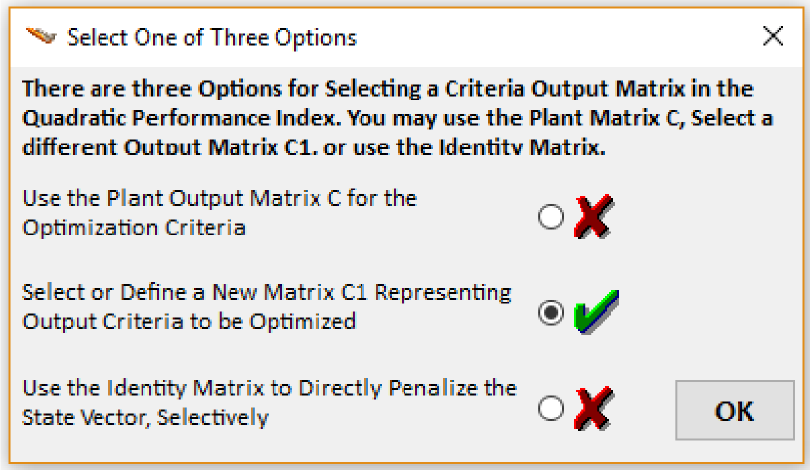




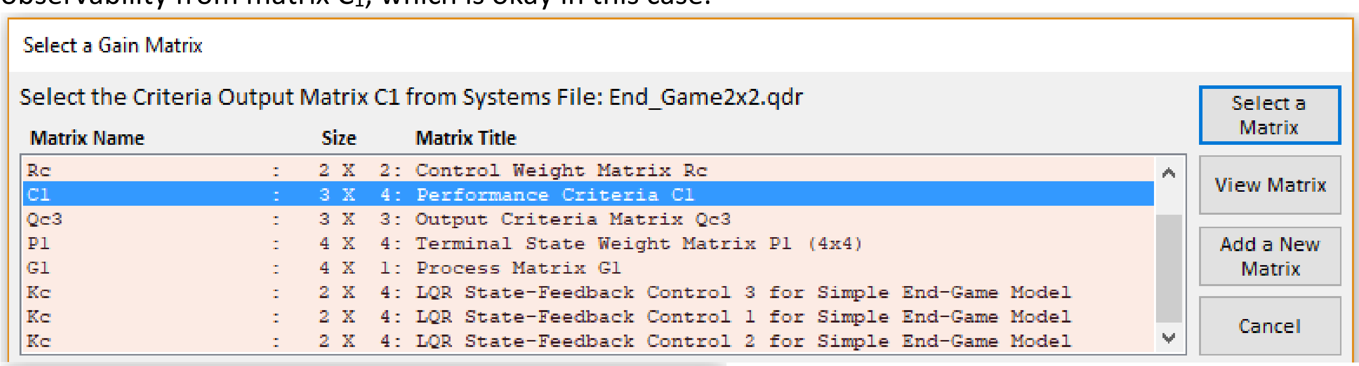
The new LQR design dataset like all datasets requires a title. Enter its title in the following dialog and click “OK”. It can be used to reprocess this operation in the future when you run the program in batch mode.

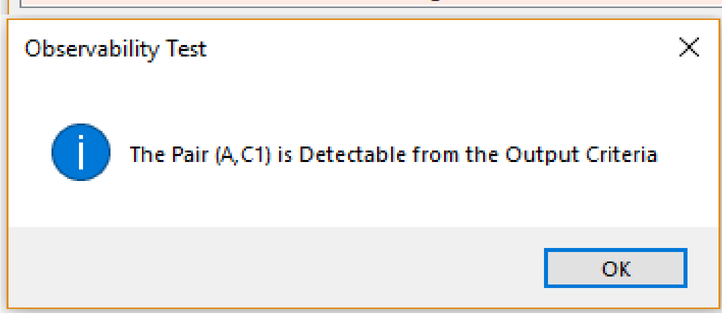


The next step is to select the output criteria to be optimized. You can either use the output matrix C, the identity matrix, or define a new set of output criteria by picking a different matrix C_1, as shown.

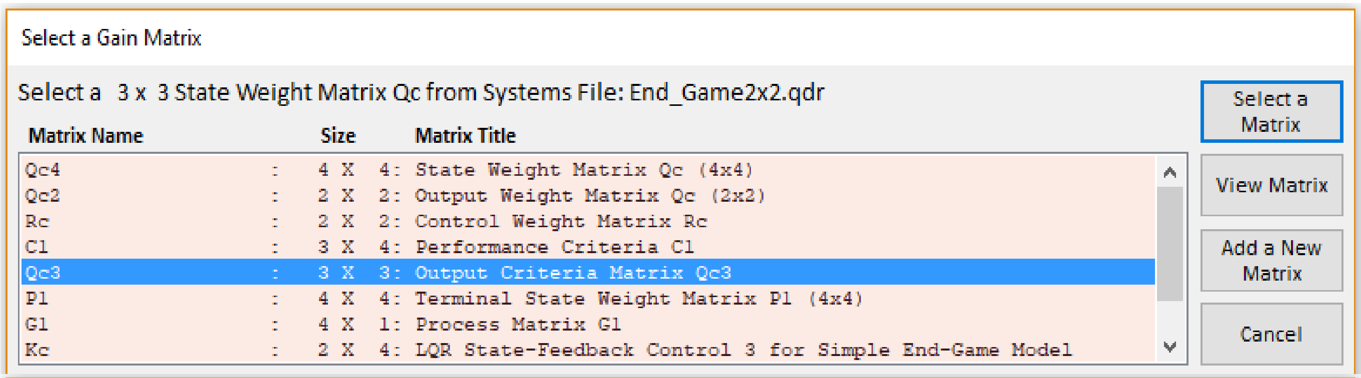


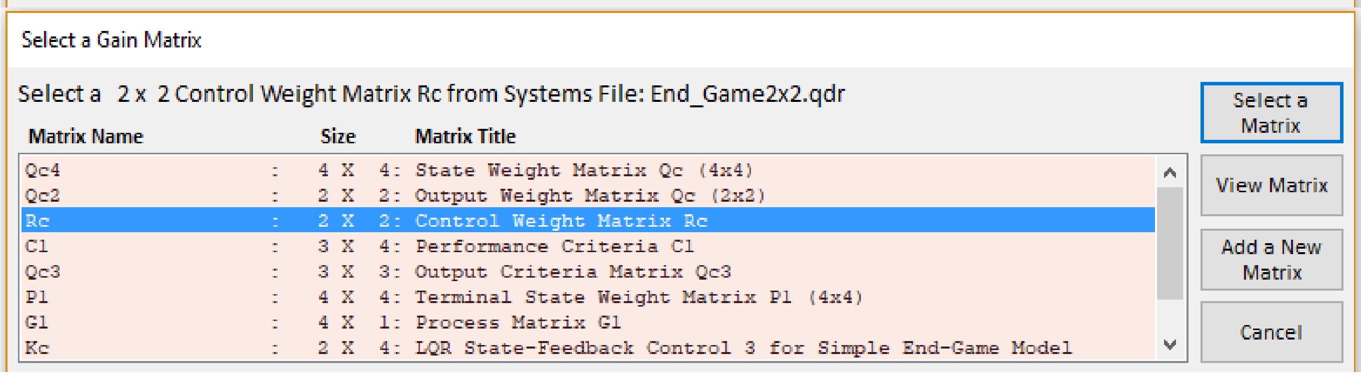
The following menu is used to select the output criteria matrix C_1 from the systems file. If the matrix is not in file you may create it interactively by adding a new matrix. The program checks the system’s observability from matrix C_1, which is okay in this case.



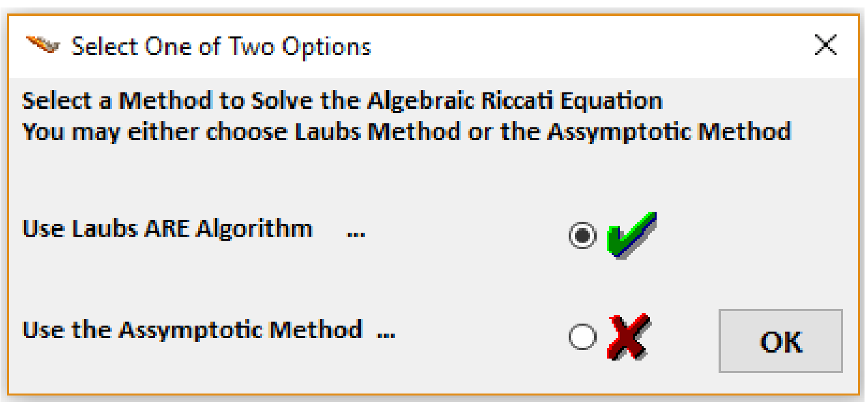


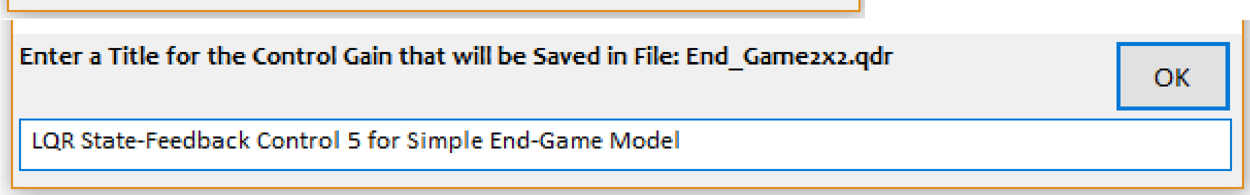
You must also select the two weight matrices Qc and Rc from the systems file. The (3x3) matrix Qc penalizes the 3 criteria outputs which are specified by the output matrix C_1, and the (2x2) matrix Rc penalizes the control inputs which are 2 in this example.





We must finally select the algorithm to solve the asymptotic Riccati equation. The program provides 2 options. Laub's algorithm is chosen in this case. We must also enter a title for the state-feedback gain Kc that will be saved in the systems file. All systems and matrices need a title in a (.Qdr) file.





The following LQR design dataset was created in the input file (.Inp) that can repeat this operation in the future. The dataset includes a label on the top: "LINEAR QUADRATIC REGULATOR …" that specifies which Flixan program will process the dataset, and a title "LQR Control Design 5 for Simple …". The green comments were added later. It can be used to reprocess the data in batch mode.

```
LINEAR QUADRATIC REGULATOR STATE-FEEDBACK CONTROL DESIGN
LQR Control Design 5 for Simple End-Game Model
! LQR Control Design for a Simple End-Game Model using
! the Criteria Optimization Outputs from Matrix C1, the
! (3x3) Output Weight Matrix Qc3, and the (2x2) Control
! weight matrix Rc. Laub's method is used to solve the ARE
!
Plant Model Used to Design the Control System from:        Simple End-Game Model
Criteria Optimization Output is Matrix C1                  Performance Criteria C1
State Penalty Weight (Qc) is Matrix:   Qc3                 Output Criteria Matrix Qc3
Control Penalty Weight (Rc) is Matrix: Rc                  Control Weight Matrix Rc
Continuous LQR Solution Using Laub Method
LQR State-Feedback Control Gain Matrix Kc                  LQR State-Feedback Control 5 for Simple End-Game
-------------------------------------------------------------------------------------------------------------
```

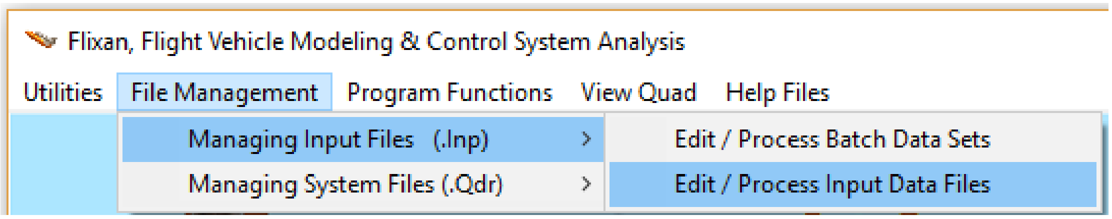
The (2x4) state-feedback gain matrix Kc was saved in the systems file (.Qdr) under the specified title. The comments are transferred from the input dataset to the matrix in the systems file. The definitions of the matrix inputs and outputs are determined from the plant model variables.

```
Gain Matrix for ...
LQR State-Feedback Control 5 for Simple End-Game Model
! LQR Control Design for a Simple End-Game Model using the Criteria Optimization
! Outputs from Matrix C1, the (3x3) Output Weight Matrix Qc3, and the (2x2) Control
! weight matrix Rc. Laub's method is used to solve the ARE
!
Matrix Kc                    Size =  2 X  4
           1-Column            2-Column            3-Column            4-Column
   1-Row -0.999999945015E+01 -0.141873048039E+02 -0.684422068284E+00  0.906398261604E+01
   2-Row  0.234488998893E-05  0.344151158201E-05  0.166220158816E-06 -0.342211034119E-08
-------------------------------------------------------------------------------------------
Definitions of Matrix Inputs (Columns):      4
Relative Position
Relative Velocity
Target Acceleration
Interceptor Acceleration

Definitions of Matrix Outputs (Rows):        2
Interceptor Acceleration Command
Target Acceleration Noise
-------------------------------------------------------------------------------------------
```

5.1 Running the Program in Batch Mode

To run a previously created dataset, such as an LQR design, for example, you must first select the project directory, and from the Flixan main menu, go to "*File Management*", "*Managing Input Files*", and then "*Edit/ Process Input Data Files*", as shown below.



The input file management utility dialog comes up and from the left menu select the input file that contains the datasets for this project by clicking on "*Select Input File*". The menu on the right fills with the titles of the datasets which are included in the input file. Select one of the titles, an LQR Control Design in this example, and click on "*Process Input Data*". The program will calculate the LQR gain matrix Kc and save it in the systems file, as before. You may then select another dataset, such as a Transient LQR or State Estimator, and process them also. If you include a batch set, such as the one shown here at the top, you may select it to instantly process the entire input file.