Cambridge University Engineering Dept. Third year

Module 3F2: Systems and Control

LECTURE NOTES 5: Putting it all together

Contents

1 Choosing L and K **2**

1.1 The Linear Quadratic Regulator 3

1.2 Kalman Filter 5

G. Vinnicombe
March 2020

1 Choosing L and K

We have seen that by using an observer based controller we can stabilize *any* plant of *degree n* (i.e. the number of states in a minimal realisation) with a controller of degree n.

Moreover, we can choose the $2n$ closed loops poles arbitralily, as

$$\{\lambda_i(A - BK)\} \cup \{\lambda_i(A - LC)\}$$

Rather than choosing these poles arbitrarily, is there a better way of choosing L and K?

1.1 The Linear Quadratic Regulator

Consider the system

$$\begin{aligned}\dot{\underline{x}} &= A\underline{x} + B\underline{u}, \quad \underline{x}(0) = \underline{x}_o \\ y &= C\underline{x}\end{aligned}$$

and the problem of choosing $\underline{u}$ to minimise

$$\int_0^\infty \underline{y}^T \underline{y} + \underline{u}^T \underline{u}\, dt$$

Solution:

If the the system is controllable, let $X = X^T$ be the unique positive definite solution to the Control Algebraic Ricatti Equation (CARE)

$$XA + A^T X + C^T C - XBB^T X = 0$$

and put

$$V(t) = \underline{x}^T(t) X \underline{x}(t) \qquad \Longrightarrow \qquad \dot{V} = \dot{\underline{x}}^T X \underline{x} + \underline{x}^T X \dot{\underline{x}}$$

So $\dot{V} + \underline{y}^T \underline{y} + \underline{u}^T \underline{u}$

$$\begin{aligned}&= (A\underline{x} + B\underline{u})^T X \underline{x} + \underline{x}^T X (A\underline{x} + B\underline{u}) + \underline{x}^T C^T C \underline{x} + \underline{u}^T \underline{u} \\ &= (\underline{u} + B^T X \underline{x})^T (\underline{u} + B^T X \underline{x}) + \underline{x}^T \underbrace{(XA + A^T X + C^T C - XBB^T X)}\, x\end{aligned}$$

Integrating both sides of this expression from $t = 0$ to ∞ gives

$$V(\infty) - \underbrace{V(0)}_{\underline{x}_0^T X \underline{x}_0} + \int_0^\infty \underline{y}^T \underline{y} + \underline{u}^T \underline{u}\, dt = \|(\underline{u} + B^T X \underline{x})\|_2^2$$

However $\underline{x}(t) \to 0$ as $t \to \infty$ (as the solution can be shown to be stabilising) so $V_\infty = 0$ and

$$\int_0^\infty \underline{y}^T \underline{y} + \underline{u}^T \underline{u}\, dt = \underbrace{\underline{x}(0)^T X \underline{x}(0)}_{\text{(indep. of } \underline{u})} + \underbrace{\|(\underline{u} + B^T X \underline{x})\|_2^2}_{= 0 \text{ if } \underline{u} = -B^T X \underline{x}(t)}$$

Thus the optimal control is

$$\underline{u} = -K\underline{x}, \quad \text{with } K = B^T X$$

1.2 Kalman Filter

Assume we have measurements of $\underline{u}(t)$ and $\underline{y}(t)$ and the model

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

What are the *smallest* noises/disturbances $\underline{d}$ and $\underline{n}$, in terms of $\int_{-\infty}^{t} \underline{d}^T \underline{d}\, dt + \int_{-\infty}^{t} \underline{n}^T \underline{n}\, dt$, which would make the measurement consistent with the model, and what is the corresponding estimate of the state?

The solution is given by *Kalman Filter* theory, which gives an
optimal trade-off between tracking d and rejecting n. The solution is a Luenberger observer with $L = YC^T$ where $Y > 0$ solves the quadratic matrix equation

$$AY + YA^T + BB^T - YC^T CY = 0$$

(if the system is observable, then it can be shown that such a solution exists, is unique, and that the resulting observer is stable).

i.e.

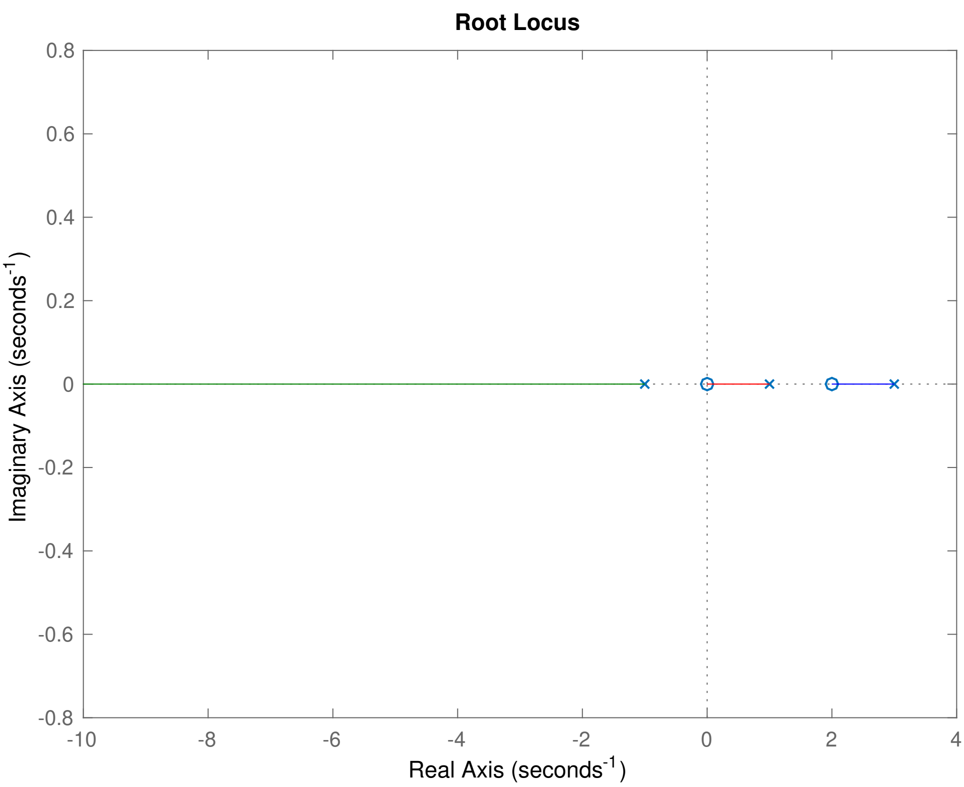
$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

This is equivalent to the *maximum likelihood* estimate under Gaussian white noise assumptions on $\underline{d}$ and $\underline{n}$ *but is is not necessary to make statistical assumptions* – the statement here just assumes that smaller noises/disturbances are more likely then larger ones.

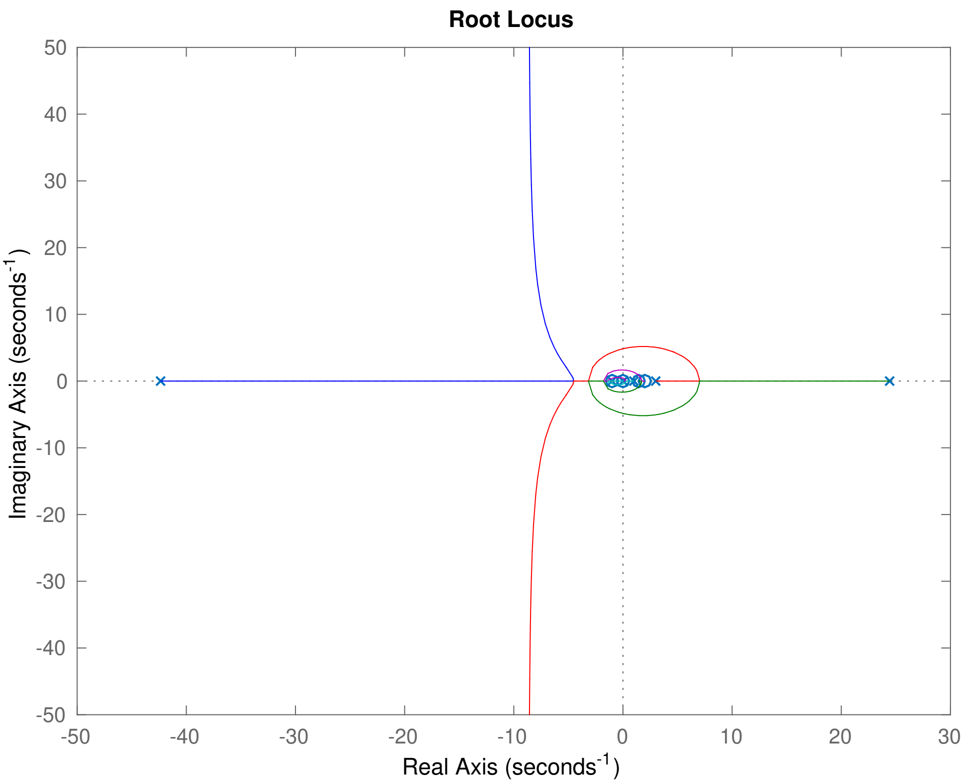
This generalises to arbitrary disturbance/noise spectra. Very widely used *Navigation & guidance, Telecomms, Control, Finance, . . .*
Especially in discrete time. Software implementation: Many (perhaps too many) in Python, in *Matlab*: `kalman, dkalman, estim` etc.

Example

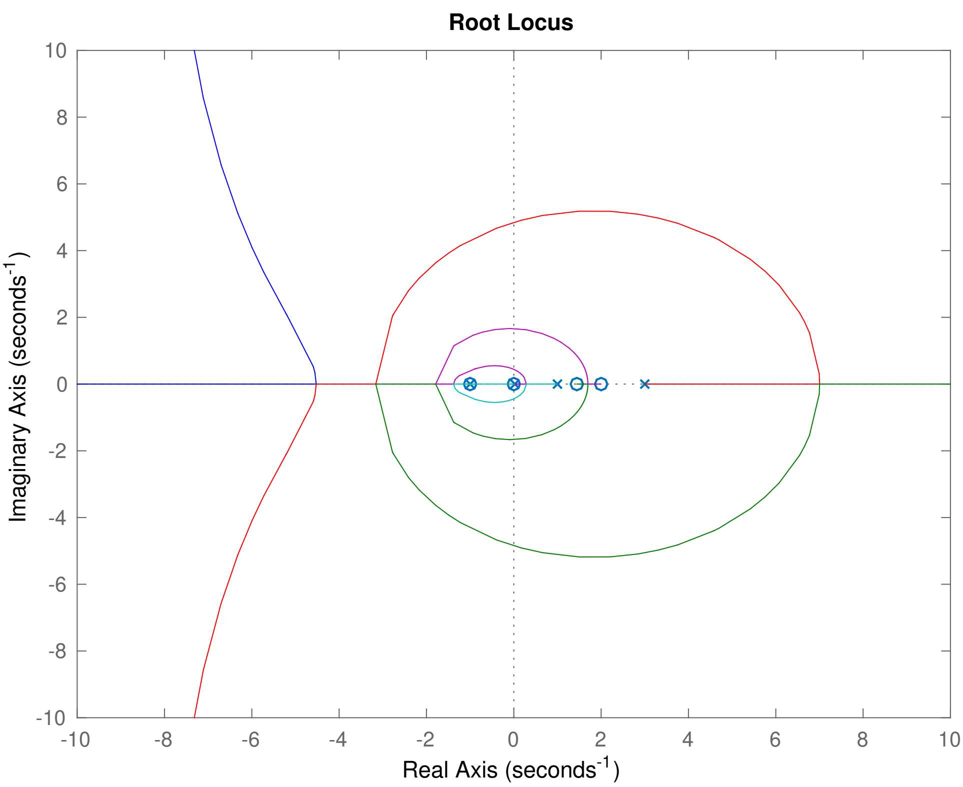
```
>> G=10*tf(poly([0 2]),poly([-1 1 3]))

G =

     10 s^2 - 20 s
  -------------------
  s^3 - 3 s^2 - s + 3

Continuous-time transfer function.

>> G=ss(G);
>> rlocus(G)
```

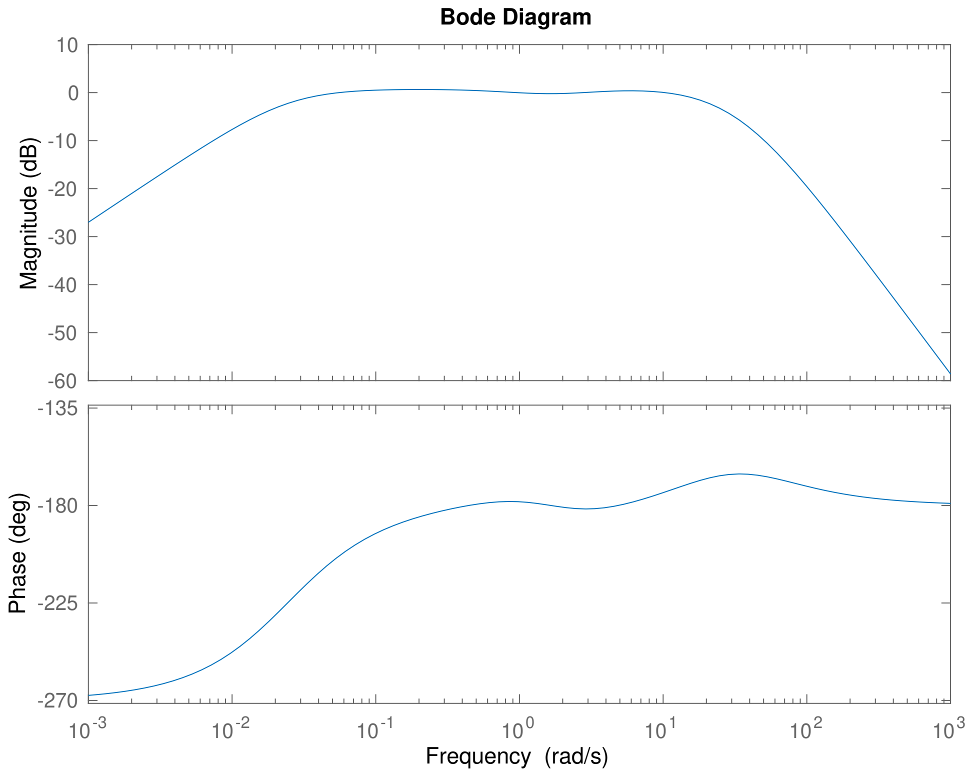


```
>> K=lqr(G.A,G.B,eye(3),1)

K =

    2.6216    1.9404    0.6930

>> L=lqr(G.A',G.C',eye(3),1)'

L =

   25.7829
   21.6166
   11.5957

>> k=ss(G.A-G.B*K-L*G.C,L,K,0)
```

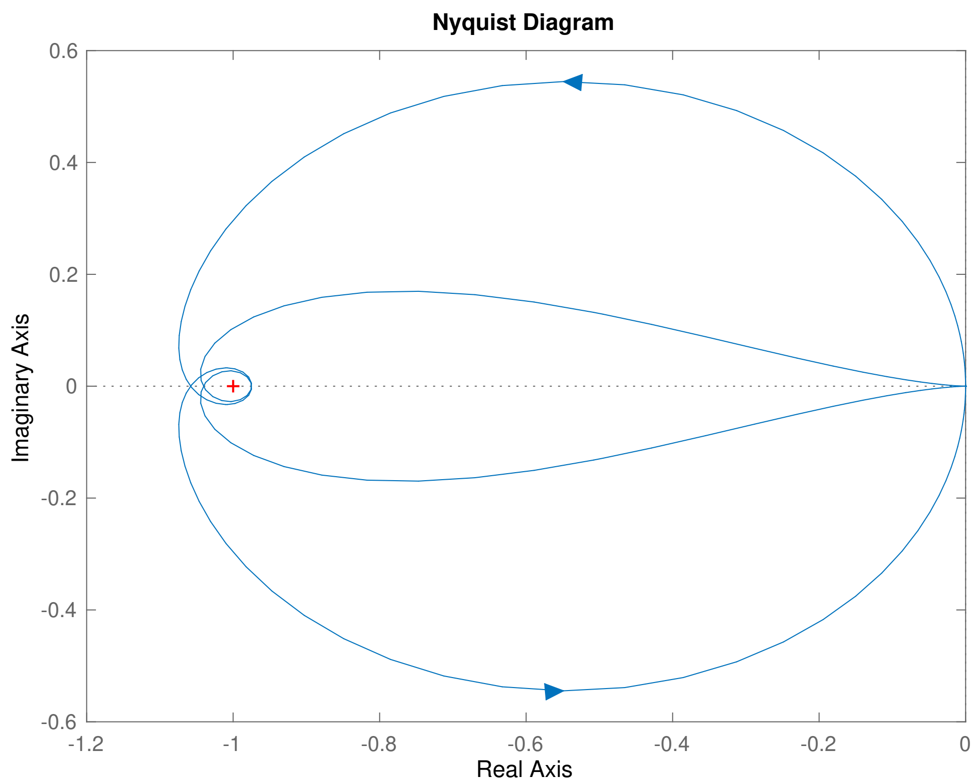


```
>> pole(feedback(G,k))
ans =
  -4.8597 + 0.0000i
  -4.1151 + 0.0000i
  -2.3983 + 0.0000i
  -1.3134 + 0.1821i
  -1.3134 - 0.1821i
  -0.9024 + 0.0000i
```





Bode Diagram
Magnitude (dB)
10
0
-10
-20
-30
-40
-50
-60
Phase (deg)
-135
-180
-225
-270
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
Frequency (rad/s)



Nyquist Diagram
Imaginary Axis
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-1.2
-1
-0.8
-0.6
-0.4
-0.2
0
Real Axis

3F2: A SUBSPACE ODDITY