16-642 Fall 2017: Reference Notes for Linear State Space Control Systems

George Kantor

24-30 October 2017

1 Quick Review

In the last class we introduced nonlinear state space systems:

We also introduced the concept of and equilibrium point: a point x_e is an equilibrium point if $\dot{x}$ is zero at x_e for the unforced system, in other words

$$f(x_e, 0) = 0.$$

In intuitively, if the system starts at an equilibrium point and is not pushed it will stay there forever!

Finally, we formally defined notions of stability of equilibrium points:

- stable means that if you start close enough to the equilibrium point, then you will stay close to it forever.
- asymptotically stable means if you start close enough to the equilibruim point, then you will converge to it.
- unstable means not stable, or that there are places arbitrarily close to the equilibrium point such that if you start there, you will go away from the equilibruim point.

2 Linearization

It is usually possible to approximate a nonlinear system with a linear system near one of its equilibrium points (nonlinear systems, unlike linear systems, can have multiple distinct equilibrium points). Here we show how to "linearize", i.e., find a linear approximation of a nonlinear system about a general equilibrium point.

2.1 Taylor Series

First, we need to revisit grade school to remind ourselves about the Taylor series and the approximations that can result from it.

Consider a function $g : \mathbb{R} \to \mathbb{R}$, $g : x \mapsto g(x)$. The Taylor series says that

$$g(x_0 + \epsilon) = g(x_0) + \frac{dg}{dx}(x_0)\epsilon + \frac{d^2g}{dx^2}(x_0)\frac{\epsilon^2}{2} + \frac{d^3g}{dx^3}(x_0)\frac{\epsilon^3}{3!} + \cdots$$

This can also be written as

$$g(x_0+\epsilon)=\sum_{k=0}^{\infty}\frac{d^kg}{dx^k}(x_0)\frac{\epsilon^k}{k!}.$$

When ϵ is small, then the series is dominated by the lower order terms. This leads to the idea of approximating g near x_0 with a truncated Taylor series. The nth order Taylor approximation of g is then

$$g(x_0+\epsilon)=\sum_{k=0}^{n}\frac{d^kg}{dx^k}(x_0)\frac{\epsilon^k}{k!}.$$

The one we are particularly interested in is the first order approximation:

$$g(x_0+\epsilon)=g(x_0)+\frac{dg}{dx}(x_0)\epsilon.$$

Of course, we are going to be applying this approximation to a vector-valued function f in order to linearize our nonlinear system. So we need to do a few things to get the answer we need out of this last equation:

1. we need the $g(x_0)$ term to disappear. This is why we linearize about equilibrium points. By definition, $f(x_e,0)=0$ at an equilibrium point.
2. we need x_0 to also be zero, which (when combined with 1.) makes the last equation look like

   $$g(\epsilon)=\frac{dg}{dx}(x_0)\epsilon.$$

3. we need to generalize Taylor series to work for vector valued function with multiple inputs. In particular, we need to generalize the notion of the derivative.

We won't go into the details of these things, we will just state how linearization works for nonlinear systems in the following section. Hopefully you will be able to make the connection.

2.2 Back to Nonlinear Systems

Consider the nonlinear system

$$\dot{x}=f(x,u)$$

$$y=h(x)$$

that has an equilibrium point x_e. Here are the steps to find a linear approximation:

1. define new coordinates such that the x_e becomes zero in the new coordinates:

   $$z=x-x_e$$

2. rewrite f and h in these new coordinates

   $$\dot{z}=\dot{x}=f(x)=f(z+x_e)\triangleq f_z(z)$$

   $$y=h(z+x_e)=h_z(z)$$

   The result is a new system that has an equilibrium point at zero.

3. the linear approximation is then

$$\dot{z} \approx \left.\frac{\partial f_z}{\partial z}\right|_{z=0,u=0} z + \left.\frac{\partial f_z}{\partial u}\right|_{z=0,u=0} u$$

$$y \approx \left.\frac{\partial h_z}{\partial z}\right|_{z=0} z$$

where for any function $g(x)$, $g : \mathbb{R}^n \to \mathbb{R}^m$:

$$\frac{\partial g}{\partial x} \triangleq \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{bmatrix}$$

Thus, the system is approximated by a linear state space system around an equilibrium point:

$$\dot{z} = Az + Bu,$$

where A is an $n \times n$ matrix and B is an $n \times m$ matrix.

3 Some Background on Matrices

We won't cover this in class, but it is here if you need to brush up on some basic facts about matrices.

Matrix Invertibility: Let A be any square matrix. **TFAE** (the following are equivalent):

1. A is invertible
2. A is nonsingular
3. det$(A) \neq 0$
4. all of the eigenvalues of A are non-zero
5. all of the rows of A are linearly independent
6. all of the columns of A are linearly independent
7. $Ax = 0$ if and only if $x = 0$.

Eigenvalues and Eigenvectors: Let $A \in \mathbb{R}^{n \times n}$ and let $e \in \mathbb{R}^n$. Then Ae is also in $\mathbb{R}^n$. So it is conceivable that $\lambda \in \mathbb{R}$ exists such that

$$Ae = \lambda e$$

In fact, it is more than conceivable: there are always n (e, λ) pairs for every A. Both e and λ might be complex. Technically, there are an infinite number of pairs, because if (e, λ) solve the above equation, then $(\gamma e, \lambda)$ also solve the above equation for any $\gamma \in \mathbb{R}$. But if we only consider e's that are unit vectors (i.e., $\|e\| = 1$) then there are n.

- the collection of e's are called *(unit) eigenvectors of A*
- the collection of $\lambda's$ are callec *eigenvalues of A*

What is an eigenvalue? I'll start by talking about eigenvectors. We can think of a square matrix as a map that takes an n dimensional vector as input and returns an n dimensional vector as an output, You put in a vector $x \in \mathbb{R}^n$ and a new vector $Ax \in \mathbb{R}^n$ comes out. For most values of x, the vector Ax is completely different from x; the two vectors have different magnitudes and point in a different directions. But for a very special subset of the possible x's, the vector Ax points in the same direction as x and differs only in magnitude. For these vectors, multiplication by the matrix A is the same as multiplication by a scalar λ. These special vectors are called "eigenvectors", and they're often denoted with an e (not to be confused with the exponential!). The scalar λ is called the eigenvalue, and it is associated with the eigenvector e.

$$Ae = \lambda e$$

So how do you find the eigenvalues?

$$Ae = \lambda e$$

is the same as

$$(\lambda I - A)e = 0$$

Using the invertibility facts above, we see that a solution can only exist if the matrix $(\lambda I - A)$ is not invertible, so we can solve for λ using the equation

$$\det(\lambda I - A) = 0$$

You can also use the `eig` command in MATLAB.

4 Linear State Space Systems

Linear state space systems are a special case of nonlinear systems. We'll get back to nonlinear systems a few lectures from now, but first we will focus on systems that are exactly linear. Linear systems are important because (1) there are many standard tools for doing anaylsis, control, and estimation for them and (2) almost any system can be reasonably approximated as a linear system when operating near an equilibrium point (i.e., by linearization).

We start by defining a state vector $x \in \mathbb{R}^n$, i.e.,

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}.$$

We also define an input vector $u \in \mathbb{R}^m$ and output vector $y \in \mathbb{R}^p$:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix}, \qquad y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{bmatrix}.$$

Note that there can be more than one input and one ouput. A linear state space is represented by the following equations:

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

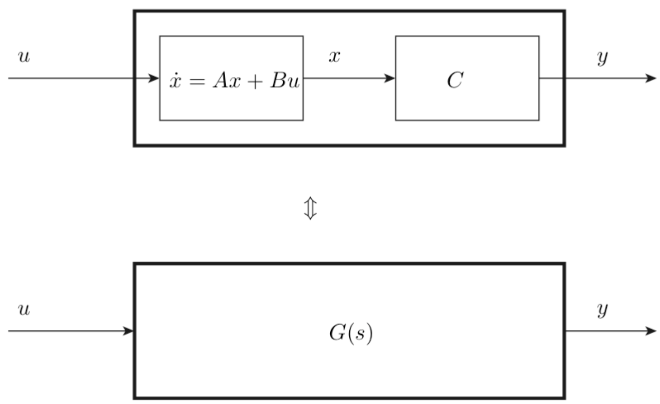
$$y(t) = Cx(t).$$

where the matrix $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. The first equation is called the state equation and the second equation is called the output equation. Sometimes you will see the output equation written as

$$y(t) = Cx(t) + Du(t).$$

This is a more general form of the state space representation. We will only deal with systems that have $D = 0$.

If the matrices (A, B, C) are constant, then the system is *linear time invariant*, or LTI for short. In this section of the class, we will deal exclusively with LTI systems. State space is just one of many ways of representing LTI systems.



5 Unforced System

It turns out that when we set $u = 0$ we can learn a lot about the behavior of a state space system. We call this case the "unforced" system:

$$\dot{x}(t) = Ax(t), \qquad x(0) = x_0,$$

5.1 Equilibrium Points

An equilibrium point is defined to be any state such that if the initial condition is that state, then the solution to the state space ODE remains at that state for all time. The only way this can happen is if $\dot{x}$ is zero, so we can mathematically define an equilibrium point x_e to be any point that satisfies:

$$\dot{x} = Ax_e = 0.$$

This means three important things for linear systems:

1. There is always an equilibrium point at $x = 0$.
2. If A is nonsingular, then the equilibrium point at $x = 0$ is unique.
3. If $x_e \neq 0$ is an equlibrium point, then γx_e is also an equilibrium point for any $\gamma \in \mathbb{R}$.

5.2 Solution

The solution to an unforced state space system is

$$x(t) = e^{At}x_0,$$

where the matrix exponential e^{At} is defined by the infinite series

$$e^{At} = I + At + \frac{(At)^2}{2} + \frac{(At)^3}{3!} + \cdots .$$

Here I is the $n \times n$ identity matrix. Using MATLAB, we can compute the matrix exponential with the `expm` command.

6 Stability Criteria

Let x_e be the equilibrium point of the linear state space system

$$\dot{x}(t) = Ax(t), \quad x(0) = 0$$

and let $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be the eigenvalues of A.

The equilibrium point $x_e = 0$ is:

- globally asymptotically stable if and only if $Re(\lambda_i) < 0 \quad \forall i$.
- unstable if and only if $Re(\lambda_i) > 0$ for some i.
- if A has no repeated eigenvalues with zero real part, then $x_e = 0$ is stable if and only if $Re(\lambda_i) \leq 0 \quad \forall i$.
- more generally, $x_e = 0$ is stable if and only if both of the following hold:
  - $Re(\lambda_i) \leq 0$ for all i
  - every eigenvalue with $Re(\lambda_i) = 0$ has an associated Jordan block of order 1.

Jordan blocks are beyond the scope of this class, so you are only responsible for the first stability result. Use the second (stronger) result if you already know what a Jordan block is or if you feel like looking it up (Google Jordan normal form).

7 Some Examples

7.1 Mass-spring-damper

The second order LTI ODE, which you could get from applying Newton's law or following the Euler-Lagrange process:

$$m\ddot{y} + \gamma\dot{y} + k_s y = u,$$

where y is the position of the mass, m is the mass, γ is the viscous friction coefficient, k_s is the linear spring constant, and u is the input force applied to the mass. Now lets get the state space representation. Define

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

then

$$\dot{x} = \begin{bmatrix} x_2 \\ -\frac{k_s}{m}x_1 - \frac{\gamma}{m}x_2 + \frac{1}{m}u \end{bmatrix}$$

which can be written in (A, B, C) form as

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & -\frac{\gamma}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

To check for stability, we find the eigenvalues of A, i.e., we look for the values of λ such that

$$\det(\lambda I - A) = 0$$

Let's do it:

$$\det(\lambda I - A) = \det\left(\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & -\frac{\gamma}{m} \end{bmatrix}\right)$$

$$= \det\left(\begin{bmatrix} \lambda & -1 \\ \frac{k_s}{m} & \lambda + \frac{\gamma}{m} \end{bmatrix}\right) = \lambda\left(\lambda + \frac{\gamma}{m}\right) - \left(-\frac{k_s}{m}\right)$$

$$= \lambda^2 + \frac{\gamma}{m}\lambda + \frac{k_s}{m}$$

so the eigenvalues of A are the solutions to the equation

$$\lambda^2 + \frac{\gamma}{m}\lambda + \frac{k_s}{m} = 0$$

which are the same as the solutions of the equation

$$m\lambda^2 + \gamma\lambda + k_s = 0.$$

We can solve for λ using the quadratic formula:

$$\lambda = \frac{-\gamma \pm \sqrt{\gamma^2 - 4mk_s}}{2m}.$$

From this we can see that the eigenvalues take different forms depending on the sign of the term under the radical. If $4mk_s > \gamma^2$ (i.e., if the damping coefficient is small compared to the mass and the spring constant) then the eigenvalues will be a complex pair with real part $-\frac{\gamma}{2m}$. Assuming that γ and m are both positive (which is almost always true for real physical systems) then the real parts of the eigenvalues are negative and the equilibrium point at zero is asymptotically stable. In the case where $\gamma > 4mks$, then there are two real eigenvalues. In this case, we can reason that the square root term is always less than γ (assuming m and k_s are positive) so again both eigenvalues are negative and the systems is asymptotically stable.

7.2 Mass-damper (no spring)

This is the same as above with $k_s = 0$, so the eigenvalues are the roots of

$$m\lambda^2 + \gamma\lambda = 0$$

which yields

$$\lambda_1 = 0, \quad \lambda_2 = -\frac{\gamma}{m}$$

Here we have all eigenvalues less then or equal to zero, and the eigenvalue with zero real part is not repeated, so we can conclude that this system is stable (and has a continuum of equilibrium points!)

7.3 Double integrator

Consider an unforced double integrator:

$$\ddot{y} = 0$$

Defining

$$x = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

we get

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x$$

so the eigenvalues of A are both zero. Since they are repeated, we cannot use the weaker stability result. In fact, this system is unstable, which we can see just by imagining how the solution will evolve for an initial condition with $x_2 \neq 0$. (We could also use the stronger stability result if we knew what a Jordan block was).

8 Controllability

A continuous time state space system is said to be *controllable* if for any two states x_0 and x_f, there exists a control signal $u(t)$, $t \in [0, t_f]$ such that if $x(0) = x_0$, then $x(t_f) = x_f$.

Loosely speaking, a system is controllable if you can get from any state to any other state.

8.1 Controllability Test

A continuous state space is controllable if and only if the matrix

$$Q = [\, B \,|\, AB \,|\, A^2B \,|\, \cdots \,|\, A^{n-1}B \,]$$

has rank n (i.e., Q has n linearly independent columns). Equivalently, a system is controllable iff $\det(QQ^T) \neq 0$.

Since controllability only depends on A and B, we sometimes say "the pair (A, B) is contollable".

9 Linear State Feedback

OK, forget about the output for awhile, and lets only worry about the state equation:

$$\dot{x} = Ax + Bu$$

The goal of linear state feedback is to find a control law

$$u = -Kx,$$

$K \in \mathbb{R}^{m \times n}$, so that the system behaves in some "nice" way. Note that under this feedback law, the state equation becomes

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

so we can understand how the closed loop system behaves using the eigenvalues of the matrix $A - BK$.

eigenvalue Placement Theorem: Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be an *allowable* set of eigenvalues, i.e., if $\lambda_i = a + bi$, $b \neq 0$, is in Λ, then $\lambda_j = a - bi$ must also be in Λ. Then the pair (A, B) is controllable if and only if there exists a $K \in \mathbb{R}^{m \times n}$ such that

$$\operatorname{eig}(A - BK) = \Lambda.$$

This is **huge!!!**. It means that I can find a gain matrix K to place the eigenvalues of a controllable system anywhere I want to!

9.1 Example: mass-spring-damper

$$m\ddot{y} + \gamma\dot{y} + k_s y = u$$

we've already seen that this becomes

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & -\frac{\gamma}{m} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x$$

Is the system controllable?

$$W = \begin{bmatrix} B & AB \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & -\frac{\gamma}{m} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \frac{1}{m} \\ \frac{1}{m} & -\frac{\gamma}{m^2} \end{bmatrix}$$

We can check the rank of W by looking at its determinant:

$$\det(W) = -\frac{1}{m^2} \neq 0$$

so W is full rank and the pair (A, B) is controllable.

Let $\Lambda = \{\lambda_1, \lambda_2\}$ be an allowable pair of eigenvalues.

The controller design problem is then to find a matrix

$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

such that the eigenvalues of $A - BK$ are λ_1 and λ_2. The eigenvalue placement theorem tells us we can do it, but it doesn't tell us how! We can figure it out though because we are smart.

$$A - BK = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m} & -\frac{\gamma}{m} \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ \frac{-k_s - k_1}{m} & \frac{-\gamma - k_2}{m} \end{bmatrix}$$

The eigenvalues of $A - BK$ are

$$\det(\lambda I - (A - BK)) = 0,$$

i.e.,

$$\Lambda = \text{roots}(\det(\lambda I - (A - BK))).$$

Subbing in our values:

$$\begin{aligned}
\Lambda &= \text{roots}\left(\det\left(\begin{bmatrix} \lambda & -1 \\ \frac{k_s+k_1}{m} & \lambda + \frac{\gamma+k_2}{m} \end{bmatrix}\right)\right) \\
&= \text{roots}\left(\lambda\left(\lambda + \frac{\gamma+k_2}{m}\right) + \frac{k_s+k_1}{m}\right) \\
&= \text{roots}\left(\lambda^2 + \lambda\frac{\gamma+k_2}{m} + \frac{k_s+k_1}{m}\right)
\end{aligned}$$

Using the quadratic formula, this becomes

$$\Lambda = \frac{-\frac{\gamma+k_2}{m} \pm \sqrt{\left(\frac{\gamma+k_2}{m}\right)^2 - 4\frac{k_s+k_1}{m}}}{2}$$

If the desired eigenvalues are complex, $\Lambda = a \pm bi$, then we can solve as follows:

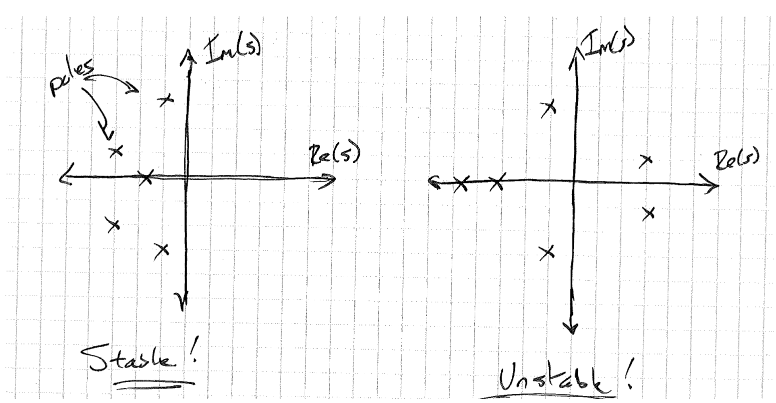
$$a = -\frac{\gamma+k_2}{2m} \quad \rightarrow \quad \boxed{k_2 = -2ma - \gamma}$$

imaginary part: it is tedious, but it can be done!

MATLAB is our friend: Fortunately, the MATLAB controls toolbox provides some eigenvalue placement functions: `place` and `acker`.

10 Complex Plane Intuition

It is useful to look at the locations of the eigenvalues of a system in the complex plane. Right now, we can look at the eigenvalue locations and quickly tell whether or not the system is stable: the system is stable if all of the eigenvalues lie in the left half plane. If any of the eigenvalues lie in the right half plane or on the imaginary axis, then the system is unstable.



10.1 Second Order Systems

In order to get an intuitive understanding how systems respond as a function of eigenvalue placement, it can be useful to study the mass-spring-damper system. After working through the example (see Section 8.1 in Lecture 7) we determined that the eigenvalues of the system are:

$$\lambda = \frac{-\gamma \pm \sqrt{\gamma^2 - 4mk_s}}{2m}.$$

In the "normal" case, $\gamma > 0$ and $k_s > 0$, and we have good intuition for how we expect the system to behave. This case can be broken down even further into the following three cases:

case 1 (underdamped): in this case, the damping is small, which we quantify by the condition $\gamma^2 < 4mk_s$. This means that the sign of the term under the square root is negative, which means that the eigenvalues are complex. The real part is $-\gamma/2m$, so the system is asymptotically stable. As γ increases, the real parts of the eigenvalues are pushed further in the left half plane, and the system converges faster. As the spring gets stiffer (i.e., as k_s gets bigger), the magnitude of the imaginary parts of the eigenvalues increases and the frequency of oscillation increases.

These two observations can be generalize to apply to all stable second order systems with complex eigenvalues: as the real part gets more negative, the rate of convergence increases; and as the imaginary part gets bigger, the frequency of oscillation increases.

case 2 (critically damped): in this case, $\gamma^2 = 4mk_s$, which means that there are two eigenvalues on top of each other at $-\gamma/2m$. In this case, there is no oscillatory component to the system response. The system will converge directly to the origin without any overshoot. As in case 1, the rate of convergence will increase this pair of eigenvalues is pushed leftward in the complex plane.

case 3 (overdamped): in this case, damping is large, which we quantify by the condition $\gamma^2 > 4mk_s$. Here the sign of the term under the square root is positive, which means the two eigenvalues are separate and real. As with the critically damped case, there is no oscillatory component to the system response. The rate of convergence is determined by the rightmost eigenvalue. As the damping gets larger, this eigenvalue moves rightward along the real axis, making the rate of convergence slower. Note that the square root term is guaranteed to be less than γ, which means that the rightmost eigenvalue is guaranteed to be negative, which means that the system is always stable.

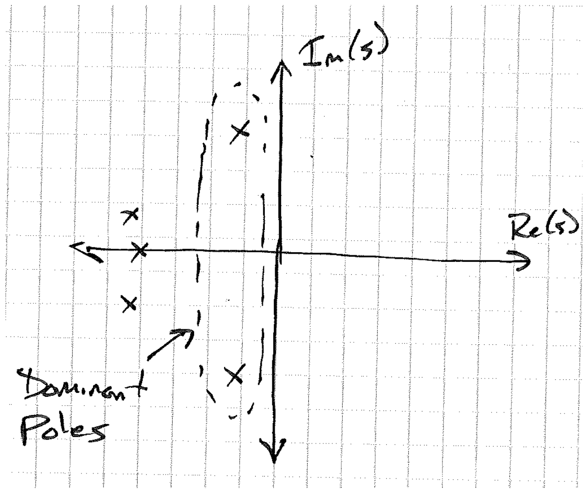
The non-normal cases are less physically meaningful, but still give good intuition and are fun to think about.

First, consider the case of negative damping. This means that there is a force that pushes the mass in the direction of its current velocity. The faster the mass moves, the harder the force pushes. Intuitively, this will obviously lead to instability. And the math matches: since γ is negative, the real parts of the eigenvalues will be positive in all cases, and the system is indeed unstable.

Next, consider the case of a negative spring constant. This means that the spring pushes the mass away from zero, and the further away the mass is the harder it pushes. Again, this intuitively will lead to instability. And the math matches. Since $4mk_s$ is negative, the term under the square root will be positive and greater than γ^2, which means the numerator will always yield one positive and one negative eigenvalue, which means the system is unstable.

10.2 Dominant Eigenvalues

If there is a set of eigenvalues that are much closer to the imaginary axis than the other eigenvalues, those eigenvalues will dominate the response. It is not unusual for one or two eigenvalues to be dominant, allowing a higher order system to be approximated by a first or second order system. This is part of the reason why it is important to thoroughly understand second orders systems: many higher order systems act like second order systems!



11 Linear Quadratic Regulator

We know a lot about eigenvalues, and it is nice to be able to place them wherever we want. However, it can be overwhelming to figure out the "best" place to put the eigenvalues, especially when the order to the system is large. Optimal control theory can help simplify things, or at least put them into a more intuitive framework.

The basic idea of any optimal control strategy is to

1. define a cost function $J : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, where $\mathbb{X}$ and $\mathbb{U}$ represent the set of continuous functions of time in the state and input spaces, respectively.
2. find a feedback control law $u = k(x)$ that minimizes J while possibly also satisfying some constraints (such as $x(t_f) = x_f$.

If you can solve the above problem, then instead of directly placing eigenvalues, we can design the controller by adjusting the parameters of J.

Let's make that more concrete, and look at an optimal control problem called *infinite time horizon linear quadradic regulation*, or just LQR for short.

LQR problem statement:
Define the cost function

$$J(x(t), u(t)) = \int_0^\infty (x(t)^T Q x(t) + u(t)^T R u(t)) dt$$

Where Q and R are

1. **symmetric**, i.e. $Q = Q^T$ or $q_{ij} = q_{ji}$
2. **positive definite**, i.e. $x^T Q x > 0$ for any $x \in \mathbb{R}^n$ and $u^T R u > 0$ for any $u \in \mathbb{R}^m$. Alternatively, all of the eigenvalues of Q and R are positive.

Now find the state feedback matrix K^* so that the closed loop system under the feedback law $u = -K^* x$ results in a lower value of J than any other choice of K.

What does this mean intuitively? To simplify lets examine the case where $n = m = 1$. Then J becomes

$$J(x,u) = \int_0^\infty (Qx(t)^2 + Ru(t)^2)dt$$

Now we can use this cost function to evaluate "how good" a feedback controller is. Assume that the feedback law being considered is $u = -Kx$, i.e., linear state feedback for some matrix K, and further assume that the resulting closed loop system is asymptotically stable. This means that both x and u go to zero at $t \to \infty$. The cost function integrates the weighted squares of these functions, capturing both some combination of how fast the functions go to zero and how large they are in the mean time.

Now, more generally, we can think of

1. $\int x^T Qxdt$ is a penalty for not having $x = 0$, i.e., penalty for not getting to the goal.
2. $\int u^T Rudt$ is a penalty for not having $u = 0$, i.e., penalty for using input energy ("price of gas").

Fortunately, the problem of finding the linear feedback matrix K that minimizes J when the pair (A, B) is controllable has already been solved.

And guess what? MATLAB has a command that solves it! Namely

$$K = \texttt{lqr}(A, B, Q, R)$$

Gives the K that minimizes J.

Now we can think about controller design in terms of a tradeoff between Q and R.

- making Q larger will drive x to zero faster.
- making R larger will reduce the control energy expended.

LQR feedback law for the double mass-spring example can be found in `lec16_example.m`.

12 Quick Review

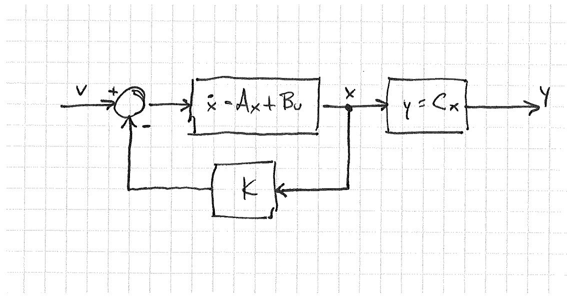
So we have two extremely powerfull tools for doing state feedback that are guaranteed to work as long as (A, B) is controllable:

- *Eigenvalue placement*: We can find a K to put the eigenvalues of the the closed loop system into any desired set of locations. Use the matlab `place` command.
- *linear quadratic regulator (LQR)*: We can get good closed loop performance by defining the symmetric positive-definite matrices Q (i.e., cost for being away from goal) and R (i.e., the "price of gas"). LQR finds the feedback matrix K so that the closed loop system optimizes the integral of the sum of the two cost components. Use the matlab `lqr` command.

These tools assume we can directly measure the state, which is not always true. So what do you do when the state cannot directly be measured? We'll get to that later today. But first, let's look at how we can use state-feedback to create a controller that causes the output to track a reference input (like PID did).

13 State-Space Tracking Controllers

Suppose that we add the ability to include an external input v into our state feedback controller by making $u = v - Kx$, resulting in the following block diagram:



Further, assume that v is constant. What happens? The addition of v does not affect the transient response of the system, that is determined by the eigenvalues, which are unchanged by adding v. But the equilibrium point that the state will converge to (assuming the closed loop system is asymptotically stable) will change. Before adding v, the equilibrium point was at zero. We can find the equilibrium point with v added by solving for the x where $\dot{x}$ is zero, i.e.:

$$\dot{x} = 0 = Ax_e - BKx_e + Bv,$$

which gives

$$x_e = -(A - BK)^{-1}Bv.$$

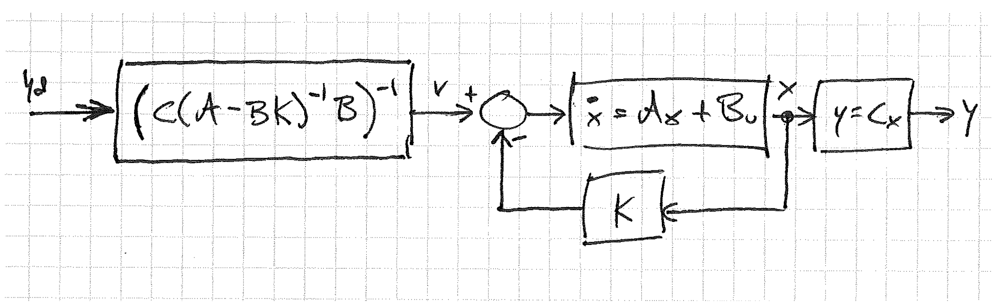
The corresponding equilibrium output is then

$$y_e = Cx_e = -C(A - BK)^{-1}Bv.$$

So, for some desired output value y_d, we can set

$$v = -\left(C(A - BK)^{-1}B\right)^{-1} y_d,$$

which will cause the output of the system to converge to $y_e = y_d$. We can add this into the block diagram as follows:



For constant y_d, we are guaranteed that $y \to y_d$, hence we can "track" constant desired outputs. We can generalize this to varying y_d signals that vary much slower than the transients in the closed-loop system. We loose any kind of guarantee, but when y_d varies slowly, the actual y will roughly track it.