EECS 16B Designing Information Devices and Systems II
Fall 2019 Note: CCF

1 Overview

Previously, we introduced the concept of stability, and explored how in some simple cases we can choose a closed-loop control rule that would stabilize an otherwise unstable system. In particular, for a system described by

$$\vec{x}(t+1) = A\vec{x}(t) + B\vec{u}(t) + \vec{w}(t),$$

we demonstrated that by making our inputs linearly depend on the state as $\vec{u}(t) = K\vec{x}(t)$, our system's state equation becomes

$$\vec{x}(t+1) = (A+BK)\vec{x}(t) + \vec{w}(t).$$

By appropriately choosing K, we can sometimes drive the eigenvalues of $A+BK$ to zero, even when the eigenvalues of the original state matrix A are large.

Notice that the same story of feedback control can apply for a differential equation

$$\frac{d}{dt}\vec{x}(t) = A\vec{x}(t) + B\vec{u}(t) + \vec{w}(t),$$

becomes

$$\frac{d}{dt}\vec{x}(t) = (A+BK)\vec{x}(t) + \vec{w}(t),$$

if we apply a control input $\vec{u}(t) = K\vec{x}(t)$. So although we will use discrete-time throughout this note (because the concept of controllability is easier to understand intuitively in the discrete-time case), all the results apply to differential equations as well.

In this lecture, we will demonstrate that when our original system is controllable, we can set the eigenvalues of $A+BK$ to whatever we like, via an appropriate choice of K. In particular, an important corollary of this result is that any unstable controllable system can be made stable through an appropriate choice of input feedback.

2 Feedback

For simplicity, we will only address the problem of a single-input system. That is to say, we can treat our input matrix B as a single vector $\vec{b}$, and our input as a scalar u. Since our system is controllable, we know that the controllability matrix

$$\mathscr{C} = \begin{bmatrix} \vec{b} & A\vec{b} & A^2\vec{b} & \cdots & A^{n-1}\vec{b} \end{bmatrix}$$

is square and of full rank, where n is the dimension of our state vector. Since our input is one-dimensional, we can write the matrix K representing the dependence of our input on the state in terms of a column vector $\vec{f}$, such that

$$K = -\vec{f}^T = -\begin{bmatrix} f_0 & f_1 & \cdots & f_{n-1} \end{bmatrix},$$

where the scalar components f_i of $\vec{f}$ are known as the *feedback gains* of our input. Notice the negative sign in the relation $K = -\vec{f}^T$ - this is to emphasize that our feedback is, at least intuitively, "cancelling out" the state at any given time.

Therefore, we find that

$$A + BK = A - \vec{b}\vec{f}^T.$$

Our problem reduces to choosing an $\vec{f}$ such that $A - \vec{b}\vec{f}^T$ has a particular set of desired eigenvalues (which would allow us to make it stable).

3 Scalar Case

As we have done many times before, we will first try to solve this problem in the scalar case, before trying to generalize. The most natural scalar system is

$$x(t+1) = ax(t) + u(t) + w(t).$$

However, we saw last time that this system is extremely easy to stabilize, by applying the input $u(t) = -ax(t)$. Intuitively, this model does not capture all the characteristics of a higher-dimensional system - specifically, when working with more than one dimension, a scalar input cannot in a single timestep completely alter the evolution of the system.

Instead, we will consider the following alternative scalar model:

$$z(t+1) = a_{n-1}z(t) + a_{n-2}z(t-1) + \cdots + a_0 z(t-n+1) + u(t) + w(t),$$

where the a_j are all scalars. Notice the change in notation from x to z. Since each $z(t+1)$ no longer depends on just the immediately previous $z(t)$ and the input, but rather depends on all of the n previous states as well as the input, we use a different letter to remind ourselves of this difference in behavior.

We see that any control input at a single time step cannot drastically alter the behavior of the system, as we expect. For instance, imagine applying some arbitrary input $u(t)$ at timestep tw. While this can affect $z(t+1)$ arbitrarily, the fact that $z(t+2)$ (and subsequent states) depend not only on $z(t+1)$, but also $z(t)$, $z(t-1)$, and so on suggests that any single input cannot arbitrarily modify the evolution of the system.

Despite this slight complexity, it should still be clear that this model can still be made to become stable. By letting our input be

$$u(t) = -a_{n-1}z(t) - a_{n-2}z(t-1) - \cdots - a_0 z(t-n+1),$$

our state equation becomes

$$z(t+1) = (a_{n-1} - a_{n-1})z(t) + (a_{n-2} - a_{n-2})z(t-1) + \cdots + (a_0 - a_0)z(t-n+1) + w(t) = w(t),$$

meaning that with these controls, any disturbance cannot build up at all over time, so our error can definitely not grow in an unbounded manner.

More generally, we can always choose our $u(t)$ to make our scalar state equation become whatever we want. Specifically, for arbitrary parameters d_0 through d_{n-1}, we can set

$$u(t) = (d_{n-1} - a_{n-1})z(t) - (d_{n-2} - a_{n-2})z(t-1) - \ldots - (d_0 - a_0)z(t-n+1),$$

so our state equation becomes

$$z(t+1) = d_{n-1}z(t) + d_{n-2}z(t-1) + \ldots + d_0 z(t-n+1) + w(t).$$

Since we could set the d_i to whatever we want, we see here that we can arbitrarily adjust the evolution of our system over time by applying an input at every timestep.

4 Controllable Canonical Form

The scalar model from the previous section sounds pretty good! It has the "vector-like" behavior we expect, due to its dependence on the previous n states, but a one-dimensional input can still be used to stabilize it. However, it still has the slightly unusual behavior of depending on the past n states, not just the most recent one.

It would be nice if we could design a vector system that behaved just like our scalar case, but depended only on the most recent (vector) state. Intuitively, we can think of our scalar system as having a "memory" of sorts, since it needs to "remember" the previous n states in order to determine the next state. A natural thing to try would be to treat this "memory" itself as our vector state.

We can represent this memory as

$$\vec{z}(t) = \begin{bmatrix} z(t-n+1) \\ \vdots \\ z(t-1) \\ z(t) \end{bmatrix}.$$

We now wish to find the matrix A_z and the vector $\vec{b}$, such that

$$\vec{z}(t+1) = A_z \vec{z}(t) + \vec{b}u(t),$$

so then we'd obtain a vector system in a form we're used to with our desired properties. From the definition of $\vec{z}$ and using basic matrix multiplication, we immediately see that

$$\vec{z}(t+1) = \begin{bmatrix} z(t-n+2) \\ \vdots \\ z(t) \\ z(t+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix} \begin{bmatrix} z(t-n+1) \\ \vdots \\ z(t-1) \\ z(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t),$$

so

$$A_z = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}$$

$$b_z = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

At this stage, a natural question to ask would be: is this vector system controllable? We should expect the answer to be yes - after all, we can clearly provide inputs $\vec{u}(t)$ to arbitrarily determine the value of $z(t+1)$, so after n timesteps, we should be able to arbitrarily set values for the full state $\vec{z}(t+n)$ (since $\vec{z}(t+n)$ depends only on the scalar states between $z(t+1)$ and $z(t+n)$).

Still, though, it can't hurt to briefly forget about our scalar interpretation of this system, and instead look at the controllability matrix. Notice that we may imagine A_z as "sliding up" the components of the vector on which it is applied. In other words,

$$A_z \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ a_0 z_0 + a_1 z_1 + \ldots + a_{n-1} z_{n-1} \end{bmatrix},$$

Thus, we may express the columns of our controllability matrix $\mathscr{C}$ as

$$\begin{aligned} \vec{b} &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}^T \\ A_z \vec{b} &= \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & ? \end{bmatrix}^T \\ A_z^2 \vec{b} &= \begin{bmatrix} 0 & 0 & \cdots & 1 & ? & ? \end{bmatrix}^T \\ &\vdots \\ A_z^{n-1} \vec{b} &= \begin{bmatrix} 1 & ? & \cdots & ? & ? & ? \end{bmatrix}^T. \end{aligned}$$

Note that we are not filling in the ? elements of these columns, since we will shortly see that they do not matter. Therefore, our final controllability matrix will be as follows:

$$\mathscr{C} = \begin{bmatrix} \vec{b} & A_z\vec{b} & \cdots & A_z^{n-1}\vec{b} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & ? \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & ? \\ 0 & 1 & ? & \cdots & ? \\ 1 & ? & ? & \cdots & ? \end{bmatrix}.$$

Since $\mathscr{C}$ is a triangular matrix with obviously[1] linearly independent columns, it is clearly of full rank. Thus, our vector system is controllable, just as we expected.

Recall that we are interested not only in controllability, but in getting stability. We demonstrated earlier that our scalar system was stabilizable when supplied with input feedback, in the sense that we could give a feedback control law that would bound the maximum deviation of $z(t)$. However, our ultimate goal is to obtain feedback gains that can place the eigenvalues of our state matrix wherever we want. A step towards this is to compute the eigenvalues of A_z in the absence of input feedback, to understand how they change as we introduce feedback.

However, our standard method of computing the determinant of $A_z - \lambda I$ seems to fail here, since computing the determinant of an $n \times n$ matrix is a tedious procedure. Another option would be to perform Gaussian elimination to try and produce a zero row, but again this seems computationally tricky and error prone (though it apparently can be done!).

Instead, rather than just computing an eigenvalue, we will try to solve for an eigenvector-eigenvalue pair simultaneously. Let such a pair be $(\lambda, \vec{v}_\lambda)$, so we have $A_z\vec{v}_\lambda = \lambda\vec{v}_\lambda$. Let the components of our eigenvector be v_i, such that

$$\vec{v}_\lambda = \begin{bmatrix} v_0 & v_1 & \cdots & v_{n-1} \end{bmatrix}^T.$$

Therefore, since $\vec{v}_\lambda$ is an eigenvector,

$$A_z \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix} = \lambda \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix}$$

$$\implies \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ a_0v_0 + a_1v_1 + \ldots + a_{n-1}v_{n-1} \end{bmatrix} = \begin{bmatrix} \lambda v_0 \\ \lambda v_1 \\ \vdots \\ \lambda v_{n-2} \\ \lambda v_{n-1} \end{bmatrix}.$$

[1] Why is linear independence obvious? None of these columns are zero. There is no way to get the last as a linear combination of the first few because only the last has a 1 in the first position. And similarly working backward. It is impossible to get the second-to-last as a linear combination of the first few for the same reason. And so on.

Equating the first $n-1$ components, we find that

$$\begin{aligned} & v_1 = \lambda v_0 \\ \implies \quad & v_2 = \lambda^2 v_0 \\ \implies \quad & v_3 = \lambda^3 v_0 \\ & \vdots \\ \implies \quad & v_{n-1} = \lambda^{n-1} v_0. \end{aligned}$$

Now, by substituting these values into the equality for the last component, we obtain

$$\begin{aligned} & a_0 v_0 + a_1 v_1 + \ldots + a_{n-1} v_{n-1} = \lambda v_{n-1} \\ \implies \quad & a_0 v_0 + \lambda a_1 v_0 + \ldots + \lambda^{n-1} a_{n-1} v_0 = \lambda^n v_0 \\ \implies \quad & v_0(\lambda^n - a_{n-1}\lambda^{n-1} - \ldots - a_1 \lambda_1 - a_0) = 0. \end{aligned}$$

We've almost got an equation for λ - there's just that factor of v_0 at the front, which we can't cancel out unless we know $v_0 \neq 0$. However, notice that if $v_0 = 0$, then all the other v_i must also equal 0, so the purported eigenvector $\vec{v}_\lambda = \vec{0}$. But by definition, the zero vector can never be an eigenvector. Therefore, $v_0 \neq 0$, so we can cancel it out to obtain

$$\lambda^n - a_{n-1}\lambda^{n-1} - \cdots - a_1 \lambda_1 - a_0 = 0,$$

something that looks suspiciously similar to the characteristic polynomial for A_z — this is a polynomial of degree n with leading coefficient 1 whose roots are the eigenvalues. (In discussion, you will prove that this indeed is the characteristic polynomial itself.) Notice that this polynomial's coefficients are basically written out for us in the bottom row of A_z — if we can manipulate those coefficients arbitrarily, then we have full control over the eigenvalues of A_z.

But in the scalar case, we showed that we could choose inputs that would make our state equation have the coefficients d_i, regardless of the initial coefficients a_i. This approach should still work in the matrix form, but we should verify it nevertheless.

Before, we chose to set

$$u(t) = (d_{n-1} - a_{n-1})z(t) - (d_{n-2} - a_{n-2})z(t-1) - \ldots - (d_0 - a_0)z(t-n+1).$$

This seems a bit worrying, since we'd like $u(t)$ to linearly depend on $\vec{z}(t)$, not on all the scalar components in some arbitrary manner. But we can rearrange this equation in vector form to obtain

$$u(t) = \begin{bmatrix} d_0 - a_0 & d_1 - a_1 & \cdots & d_{n-1} - a_{n-1} \end{bmatrix} \vec{z}(t),$$

so our feedback terms are $f_j = a_j - d_j$ for all $0 \leq j < n$.

Substituting this input into our vector state equation, we obtain

$$\begin{aligned}
\vec{z}(t+1) &= A_z\vec{z}(t)+\vec{b}u(t) \\
&= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}\vec{z}(t)+\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} d_0-a_0 & d_1-a_1 & \cdots & d_{n-1}-a_{n-1} \end{bmatrix}\vec{z}(t) \\
&= \left(\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}+\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} d_0-a_0 & d_1-a_1 & \cdots & d_{n-1}-a_{n-1} \end{bmatrix}\right)\vec{z}(t) \\
&= \left(\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}+\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ d_0-a_0 & d_1-a_1 & \cdots & d_{n-1}-a_{n-1} \end{bmatrix}\right)\vec{z}(t) \\
&= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ d_0 & d_1 & d_2 & \cdots & d_{n-1} \end{bmatrix}\vec{z}(t),
\end{aligned}$$

so we have successfully chosen our feedback gains that can arbitrarily modify the eigenvalues of our closed-loop matrix of dynamics.

5 Simplifying General Controllable Systems

Let's review what we've seen so far. We have developed our earlier scalar model into a vector state equation in our standard form, known as the *controllable canonical form*. We have demonstrated that all systems with state equations of this form are stabilizable in that input feedback can be set up in such a way that we can set the closed-loop eigenvalues to whatever we want.

But all these results only apply to a very specific type of system — those which are essentially "higher-order" scalar systems "in disguise", where the "higher order" just means that that each state depends on the past n states, not only on the most recent one. We will now make the counter-intuitive assertion that *any* controllable system with a scalar input can be viewed in this form, after applying a change of coordinates. Let this controllable system be

$$\vec{x}(t+1)=A\vec{x}(t)+\vec{b}u(t)+w(t),$$

as before.

To understand this, recall that since this system is controllable, the sequence

$$\vec{b}, A\vec{b}, A^2\vec{b}, \ldots, A^{n-1}\vec{b}$$

forms a basis for the state space. We will horizontally stack the vectors in this sequence to form the matrix $G = [\vec{b}, A\vec{b}, A^2\vec{b}, \ldots, A^{n-1}\vec{b}]$, that is clearly of full rank.

Thus, the state at any time can be written as

$$\vec{x}(t) = G \begin{bmatrix} \widetilde{x}_0(t) \\ \widetilde{x}_1(t) \\ \vdots \\ \widetilde{x}_{n-1}(t) \end{bmatrix} = \sum_{j=0}^{n-1} \widetilde{x}_j(t) A^j \vec{b},$$

for some suitable constants $\widetilde{x}_j(t)$. We may obtain these constants in vector form $\widetilde{x}(t)$ by evaluating $G^{-1}\vec{x}(t)$.

Intuitively, we should expect that applying A "slides down" our representation of the state in the basis G, since the coefficient for $\vec{b}$ becomes the coefficient for $A\vec{b}$, the coefficient for $A\vec{b}$ becomes the coefficient for $A^2\vec{b}$, and so on. This behavior sounds very similar to what we saw in canonical form (except there our state kept "sliding up"), so we should investigate and see whether our system in fact behaves like that.

Applying our state equation, we find that

$$\begin{aligned}
\vec{x}(t+1) &= \left(A \sum_{j=0}^{n-1} \widetilde{x}_j(t) A^j \vec{b} \right) + \vec{b}u(t) \\
&= \left(\sum_{j=0}^{n-1} \widetilde{x}_j(t) A^{j+1} \vec{b} \right) + \vec{b}u(t) \\
&= \left(\sum_{j=1}^{n-1} \widetilde{x}_{j-1}(t) A^j \vec{b} \right) + \widetilde{x}_{n-1}(t) A^n \vec{b} + \vec{b}u(t)
\end{aligned}$$

Now, recall that $A^n\vec{b}$ must lie in the span of the vectors in our basis, so we may represent it as

$$A^n\vec{b} = \sum_{j=0}^{n-1} \alpha_j A^j \vec{b}$$

for some constants α_j.

Thus, substituting our representation for $A^n\vec{b}$ into our expression for $\vec{x}(t+1)$, we obtain

$$\begin{aligned}
\vec{x}(t+1) &= \left(\sum_{j=1}^{n-1} \widetilde{x}_{j-1}(t) A^j \vec{b} \right) + \widetilde{x}_{n-1}(t) \left(\sum_{j=0}^{n-1} \alpha_j A^j \vec{b} \right) + \vec{b}u(t) \\
&= (\alpha_0 \widetilde{x}_{n-1}(t) + u(t))\vec{b} + \sum_{j=1}^{n-1} (\widetilde{x}_{j-1}(t) + \alpha_j \widetilde{x}_{n-1}(t)) A^j \vec{b},
\end{aligned}$$

so we can express $\vec{x}(t+1)$ as a linear combination of our basis vectors as well. Writing this representation in matrix-vector form, we obtain

$$\vec{x}(t+1) = G \begin{bmatrix} u(t) + \alpha_0 \widetilde{x}_{n-1}(t) \\ x_0(t) + \alpha_1 \widetilde{x}_{n-1}(t) \\ x_1(t) + \alpha_2 \widetilde{x}_{n-1}(t) \\ \vdots \\ x_{n-2}(t) + \alpha_{n-1} \widetilde{x}_{n-1}(t) \end{bmatrix}.$$

Therefore, we may express $\vec{x}(t+1)$ in this new basis as

$$\widetilde{x}(t+1) = G^{-1}\vec{x}(t+1) = \begin{bmatrix} u(t) + \alpha_0 \widetilde{x}_{n-1}(t) \\ x_0(t) + \alpha_1 \widetilde{x}_{n-1}(t) \\ x_1(t) + \alpha_2 \widetilde{x}_{n-1}(t) \\ \vdots \\ x_{n-2}(t) + \alpha_{n-1} \widetilde{x}_{n-1}(t) \end{bmatrix} = \begin{bmatrix} \alpha_0 \widetilde{x}_{n-1}(t) \\ x_0(t) + \alpha_1 \widetilde{x}_{n-1}(t) \\ x_1(t) + \alpha_2 \widetilde{x}_{n-1}(t) \\ \vdots \\ x_{n-2}(t) + \alpha_{n-1} \widetilde{x}_{n-1}(t) \end{bmatrix} + \begin{bmatrix} u(t) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

You might notice that this equation looks suspiciously familiar to our old scalar system.

We can see these similarities more closely by rearranging this expression, to obtain

$$\begin{aligned} \widetilde{x}(t+1) &= \begin{bmatrix} \alpha_0 \widetilde{x}_{n-1}(t) \\ x_0(t) + \alpha_1 \widetilde{x}_{n-1}(t) \\ x_1(t) + \alpha_2 \widetilde{x}_{n-1}(t) \\ \vdots \\ x_{n-2}(t) + \alpha_{n-1} \widetilde{x}_{n-1}(t) \end{bmatrix} + \begin{bmatrix} u(t) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 & \cdots & \alpha_0 \\ 1 & 0 & 0 & \cdots & \alpha_1 \\ 0 & 1 & 0 & \cdots & \alpha_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{n-1} \end{bmatrix} \begin{bmatrix} \widetilde{x}_0(t) \\ \widetilde{x}_1(t) \\ \widetilde{x}_2(t) \\ \vdots \\ \widetilde{x}_{n-1}(t) \end{bmatrix} + u(t) \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= \widetilde{A}\widetilde{x}(t) + u(t)\widetilde{b}, \end{aligned}$$

where

$$\begin{aligned} \widetilde{A} &= \begin{bmatrix} 0 & 0 & 0 & \cdots & \alpha_0 \\ 1 & 0 & 0 & \cdots & \alpha_1 \\ 0 & 1 & 0 & \cdots & \alpha_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_{n-1} \end{bmatrix} \\ \widetilde{b} &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned}$$

Notice that all we have really done is performed a change of basis, since $\widetilde{x} = G^{-1}\vec{x}$. That is to say, starting with our original state equation, we have simply pre-multiplied by G^{-1}, to obtain

$$\begin{aligned} &\vec{x}(t+1) = A\vec{x}(t) + \vec{b}u(t) \\ \implies \quad &G^{-1}\vec{x}(t+1) = G^{-1}A(GG^{-1})\vec{x}(t) + G^{-1}\vec{b}u(t) \\ \implies \quad &\widetilde{x}(t+1) = (G^{-1}AG)\widetilde{x}(t) + G^{-1}\vec{b}u(t). \end{aligned}$$

Therefore, it should be clear that $\widetilde{A} = G^{-1}AG$ and $\widetilde{b} = G^{-1}\vec{b}$.

One question that could be asked at this stage is - why go through all that complicated matrix algebra, when all we were doing was pre-multiplying by G^{-1}? The reason for all that algebra was to establish the form for $\widetilde{A}$ and $\widetilde{b}$. Unlike with diagonalization, where we knew we would obtain a diagonal matrix in the eigenbasis, here we had no idea what our state matrix would look like in the new basis G, so we had to algebraically work it out.

6 Achieving Controllable Canonical Form

Unfortunately, despite all this computation, we aren't quite there yet! Our controllable canonical form, with all of its nice properties, had a *row* of constants at the bottom - but so far, we've only managed to get these constants into a column!

To resolve this, we will make the *conjecture* that there exists a controllable canonical form for our system with a *particular* state transition matrix. Specifically, we will hypothesize that the state equation:

$$\vec{z}(t+1) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} \end{bmatrix} \vec{z}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t) = \widetilde{A}^T\vec{z}(t) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t)$$

models the behavior of our system under a further change of basis $\widetilde{x} = H^{-1}\vec{z}$. We will aim to demonstrate both the correctness of this hypothesis, and to compute the change of basis matrix H itself.

In the previous section, we demonstrated that any controllable system can be rewritten in a simpler form, via a change of basis. At the moment, the system with state matrix $\widetilde{A}$ is in that simplified form, though our target state equation with state matrix $\widetilde{A}^T$ is not. Rather than trying to convert the former system into the latter, it might be easier to try and convert the latter system into the former, since we can always reverse our change of basis transformation later.

In particular, recall that the change of basis represented by the matrix

$$G = \begin{bmatrix} \vec{b} & A\vec{b} & \cdots & A^{n-1}\vec{b} \end{bmatrix}$$

converted the system involving A (an arbitrary state matrix) into a simplified form with state matrix $\widetilde{A}$.

Similarly, we know that choosing the change of basis matrix

$$H = \begin{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} & \widetilde{A}^T \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} & \cdots & (\widetilde{A}^T)^{n-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

will convert the system involving $\widetilde{A}^T$ into a simplified form. We will represent the state equation of this simplified form as

$$\vec{z}'(t+1) = A'\vec{z}' + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u(t),$$

where A' is of the same form as $\widetilde{A}$, except with potentially different constants in the last column. Let these constants be β_i, so

$$A' = H^{-1}\widetilde{A}^T H = \begin{bmatrix} 0 & 0 & 0 & \cdots & \beta_0 \\ 1 & 0 & 0 & \cdots & \beta_1 \\ 0 & 1 & 0 & \cdots & \beta_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_{n-1} \end{bmatrix}.$$

We will now aim to determine what these constants β_i are.

To do so, we will consider the characteristic polynomial of A'. Recall that the characteristic polynomial of a matrix in controllable canonical form depends only on the coefficients on the bottom row. Specifically, for the matrix $\widetilde{A}^T$ (which is clearly in that form), we know that its characteristic polynomial in reduced form is

$$|\lambda I - \widetilde{A}^T| = \lambda^n - \alpha_{n-1}\lambda^{n-1} - \ldots - \alpha_1\lambda_1 - \alpha_0.$$

Using elementary properties of the determinant, we also know that the characteristic polynomial of A' is the same as the characteristic polynomial of its transpose $(A')^T$. Since $(A')^T$ is in controllable canonical form, we know that its characteristic polynomial in reduced form is

$$|\lambda I - A'| = |\lambda I - (A')^T| = \lambda^n - \beta_{n-1}\lambda^{n-1} - \ldots - \beta_1\lambda_1 - \beta_0.$$

Why does all this matter? Well, at an intuitive level, the characteristic polynomial "captures" some sort of intrinsic property of a linear transformation - specifically, its eigenvalues and their multiplicities. More formally, we can show using elementary properties of the determinant that the characteristic polynomials of two matrices related by a change of basis (known as *similar matrices*) are the same, as follows:

$$\begin{aligned}
&& A' &= H^{-1}\widetilde{A}^T H \\
\implies && A' - \lambda I &= H^{-1}\widetilde{A}^T H - \lambda I \\
&& &= H^{-1}\widetilde{A}^T H - \lambda H^{-1} H \\
&& &= H^{-1}(\widetilde{A}^T - \lambda I) H \\
\implies && \det\left(A' - \lambda I\right) &= \det\left(H^{-1}(\widetilde{A}^T - \lambda I) H\right) \\
&& &= \det\left(H^{-1}\right)\det\left(\widetilde{A}^T - \lambda I\right)\det(H) \\
&& &= \det(H)^{-1}\det\left(\widetilde{A}^T - \lambda I\right)\det(H) \\
&& &= \det\left(\widetilde{A}^T - \lambda I\right).
\end{aligned}$$

Therefore, we have the equality

$$\lambda^n - \alpha_{n-1}\lambda^{n-1} - \ldots - \alpha_1\lambda_1 - \alpha_0 = \lambda^n - \beta_{n-1}\lambda^{n-1} - \ldots - \beta_1\lambda_1 - \beta_0,$$

for all λ. Consequently, it is clear that $\alpha_i = \beta_i$ for all valid values of i.

Looking back at the form of A', we see that $A' = \widetilde{A}$. Therefore, the state equations describing the behavior of $\widetilde{x}$ and $\vec{z}'$ are the same! In other words, given appropriate initial conditions, we have that $\widetilde{x}(t) = \vec{z}'(t)$ for all timesteps i![2]

Recall that we had defined $\widetilde{x} = G^{-1}\vec{x}$ and $\vec{z}' = H^{-1}\vec{z}$. So we can trivially relate $\vec{x}$ and $\vec{z}$ with the change of basis

$$\vec{z} = HG^{-1}\vec{x}.$$

But whereas $\vec{x}$ was governed by an arbitrary controllable state equation, the state equation describing the behavior of $\vec{z}$ was in controllable canonical form!

Therefore, we have successfully proved that *any* controllable system can be rewritten in controllable canonical form through a change of basis. But rather than considering some "magic" change of basis, we were able to derive the change of basis HG^{-1}, by first converting our system of $\vec{x}$ into an intermediate "simplified" form, then considering an unrelated system $\vec{z}$ in controllable canonical form, converting *that* into the same intermediate simplified form, and demonstrating equality.

As a consequence, we now can apply our results for controllable canonical form to arbitrary controllable systems. Specifically, we know that with an appropriate change of basis, any controllable vector system with a single scalar input can be treated as a higher-order discrete-time scalar system (where each state depended on the past n states). Thus, we can choose appropriate feedback gains to arbitrarily place the eigenvalues of our original system, so every controllable system can be stabilized through feedback!

For differential equations, the concept of "higher-order" differential equations simply refers to differential equations in which an appropriate derivative (possibly higher than the first derivative) is what is being constrained. A higher-order linear differential equation has this higher order derivative defined in terms of constant multiples of lower-order derivatives and the function itself. Our result on controllable canonical form tells us that when the matrices are controllable, all vector systems of linear differential equations driven by a scalar input are essentially just a scalar higher-order differential equation.

[2] Yes, these exclamation marks are important! This is a fantastic result!

Contributors:

- Rahul Arya.
- Anant Sahai.