Recall

\[ x(k + 1) = Ax(k) + Bu(k) \]
\[ y(k) = Cx(k) \]
\[ x(k) = A^k x(0) + [A^{k-1} B A^{k-2} B ... B] \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(k-1) \end{bmatrix} \]

For the case where there is zero input (\( u(k) = 0 \) for all \( k \)), let us look at two cases:

1. \( A \) is diagonizable.
2. \( A \) is not diagonizable.

Example: 2x2 Matrix with only 1 eigenvector

Let’s look at what happens when \( A = \begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} \).

When the only eigenvalue of \( A \) has magnitude less than one, the system will have a decaying response over time. When the eigenvalue has magnitude larger than one, the response will grow over time.

What about when \( \lambda = 1 \)?
\( A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \) and \( A^k = \begin{bmatrix} 1 & k \\ 0 & 1 \end{bmatrix} \)

So we want to design a system such that the eigenvalues lie within the unit circle.

Input

Suppose now we add an input (e.g. actuator, force, etc). What happens to the solution \( x(k) \) for different kinds of input?

Currently, if \( \lambda \) is inside the unit circle, the system is stable. If \( \lambda \) is on or outside the unit circle, the system is, at best, marginally stable. When there is added input, if the response was unstable before, it will probably still be unstable because the instability propagates through the other terms. However, we can cleverly choose \( K \) to make the response look good.
Recall that this is the closed loop system:

$$x(k+1) = (A - BK) x(k) + BKr(k)$$

$$y(k) = Cx(k)$$

Case (i)

For this case, the open loop system is a unity gain block. $y(k) = K(r(k) - y(k))$ so $y(k)(1 + K) = Kr(k)$. Thus the single input single output (SISO) system has this response: $\frac{y(k)}{r(k)} = \frac{K}{1+K}$. As $K$ gets larger and larger, the response approaches one.

Case (ii)

Now when the open loop system is a gain block with gain $A$, $\frac{y(k)}{r(k)} = \frac{KA}{1+KA}$. As $K$ approaches infinity such that $KA$ is large, the response will approach 1.
Case (iii)

Let’s add in some noise, \( n(k) \). We would like to use feedback to remove the effects of the noise.

\[
y(k) = A(n(k) + K(r(k) - y(k))) = An(k) + KAr(k) - KAy(k)
\]

So \( y(k) = \frac{A}{1 + KA}n(k) + \frac{KA}{1 + KA}r(k) \) Because we have an LTI system, we can look at the two components assuming the other is zero. You can see that as \( K \) approaches infinity, the transfer function for noise goes to zero. This means \( K \) reduces the effect of noise.

The transfer function \( \frac{y(k)}{r(k)} = \frac{KA}{1 + KA} \)

How do we choose the gain matrix \( K \)?

**Example 1**

Let’s look at this SISO system:

\[
A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = [0 \ 1], \quad K \text{ is a real number.}
\]

If \( x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \) where \( x_1(k) \) is \( x \) position and \( x_2(k) \) is \( y \) position, then the input to the system only affects the \( y \) position.

\[
x(k + 1) = Ax(k) + Bu(k) = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)
\]

\[
y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}
\]

Then \( A - BK = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} K \begin{bmatrix} 0 & 1 \end{bmatrix} \).

Confirming dimensions, we get that \( A_{CL} = A - BK = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 - K \end{bmatrix} \). Thus the gain and input only affect the \( y \)-position.

Here, \( K \) can be selected to "change only \( \lambda_2 \)". For example, if it did not lie in the unit circle, you can use feedback to change that.

**Example 2**

Example 2 is changing \( \lambda_1 \). This case is the same as above except that, here, \( C = [1 \ 0] \). Now the input only affects the \( x \)-position.
Examples 3 and 4

For example 3: $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$, $A - BK = \begin{bmatrix} \lambda_1 & 0 \\ -K & \lambda_2 \end{bmatrix}$

For example 4: $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $A - BK = \begin{bmatrix} \lambda_1 & -K \\ 0 & \lambda_2 \end{bmatrix}$

This complies with our intuition: neither $\lambda_1$ nor $\lambda_2$ gets changed. We are only sensing one variable but we are affecting the other.

Example 5

Let $A = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. This means we are using 2 channels to receive two independent pieces of information (both state variables). Based on the dimensions, K must be of the form $K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$.

Let’s check $A - BK$ for the closed loop system. We find that $A - BK = \begin{bmatrix} \lambda_1 & 0 \\ -k_1 & \lambda_2 - k_2 \end{bmatrix}$.

So what happened here? Even though we increase the dimensions of $C$ and $K$, even though we increased the number of sensors, we can still only control $\lambda_2$. 