2 Matrices and Vectors

This Section is a very concise introduction to the algebra of matrices and vectors.

2.1 Matrices

A (real) matrix of size \( m \times n \) is an array of \( mn \) real numbers arranged in \( m \) rows and \( n \) columns:

\[
A = \begin{bmatrix}
a_{11} & \cdots & a_{1n} \\
\vdots & & \vdots \\
a_{m1} & \cdots & a_{mn}
\end{bmatrix}.
\]

The \( n \times m \) matrix \( A^T \) obtained by exchanging rows and columns of \( A \) is called the transpose of \( A \). A matrix \( A \) is said to be symmetric if \( A = A^T \).

The sum of two matrices of equal size is the matrix of the entry-by-entry sums, and the scalar product of a real number \( a \) and an \( m \times n \) matrix \( A \) is the \( m \times n \) matrix of all the entries of \( A \), each multiplied by \( a \). The difference of two matrices of equal size \( A \) and \( B \) is

\[
A - B = A + (-1)B.
\]

The product of an \( m \times p \) matrix \( A \) and a \( p \times n \) matrix \( B \) is an \( m \times n \) matrix \( C \) with entries

\[
c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj}.
\]

Examples

Let

\[
A = \begin{bmatrix} 3 & 0 & -2 \\ 1 & -1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}.
\]

Then,

\[
A^T = \begin{bmatrix} 3 & 1 \\ -2 & -1 \end{bmatrix}, \quad B^T = \begin{bmatrix} 2 & 1 \\ -1 & 3 \end{bmatrix}
\]

and

\[
C = BA = \begin{bmatrix} 2 \cdot 3 + (-1) \cdot 1 & 2 \cdot 0 + (-1) \cdot (-1) & 2 \cdot (-2) + (-1) \cdot 0 \\ 1 \cdot 3 + 3 \cdot 1 & 1 \cdot 0 + 3 \cdot (-1) & 1 \cdot (-2) + 3 \cdot 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 & -4 \\ 6 & -3 & -2 \end{bmatrix}.
\]

The product \( AB \) is not defined, because \( A \) and \( B \) have incompatible sizes. Furthermore,

\[
3A = \begin{bmatrix} 3 \cdot 3 & 3 \cdot 0 & 3 \cdot (-2) \\ 3 \cdot 1 & 3 \cdot (-1) & 3 \cdot 0 \end{bmatrix} = \begin{bmatrix} 9 & 0 & -6 \\ 3 & -3 & 0 \end{bmatrix}.
\]
\[ A + C = \begin{bmatrix} 3 & 0 & -2 \\ 1 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 5 & 1 & -4 \\ 6 & -3 & -2 \end{bmatrix} = \begin{bmatrix} 3 + 5 & 0 + 1 & -2 + (-4) \\ 1 + 6 & -1 + (-3) & 0 + (-2) \end{bmatrix} = \begin{bmatrix} 8 & 1 & -6 \\ 7 & -4 & -2 \end{bmatrix} \]

and

\[ A - C = \begin{bmatrix} 3 & 0 & -2 \\ 1 & -1 & 0 \end{bmatrix} - \begin{bmatrix} 5 & 1 & -4 \\ 6 & -3 & -2 \end{bmatrix} = \begin{bmatrix} 3 - 5 & 0 - 1 & -2 - (-4) \\ 1 - 6 & -1 - (-3) & 0 - (-2) \end{bmatrix} = \begin{bmatrix} -2 & -1 & 2 \\ -5 & 2 & 2 \end{bmatrix}. \]

### 2.2 Vectors

A (real) \( n \)-dimensional vector is an \( n \)-tuple of real numbers

\[ \mathbf{v} = (v_1, \ldots, v_n). \]

There is a natural, one-to-one correspondence between \( n \)-dimensional vectors and \( n \times 1 \) matrices:

\[ (v_1, \ldots, v_n) \leftrightarrow \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}. \]

The matrix on the right is called the column vector corresponding to the vector on the left.

There is also a natural, one-to-one correspondence between \( n \)-dimensional vectors and \( 1 \times n \) matrices:

\[ (v_1, \ldots, v_n) \leftrightarrow \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}. \]

The matrix on the right is called the row vector corresponding to the vector on the left.

If \( \mathbf{a} \) is a vector, then the symbol \( \mathbf{a} \) also denotes the corresponding column vector, so that the corresponding row vector is \( \mathbf{a}^T \).

All algebraic operations on vectors are inherited from the corresponding matrix operations, when defined. In addition, the inner product of two \( n \)-dimensional vectors

\[ \mathbf{a} = (a_1, \ldots, a_n) \quad \text{and} \quad \mathbf{b} = (b_1, \ldots, b_n) \]

is the real number equal to the matrix product \( \mathbf{a}^T \mathbf{b} \). It is easy to verify that this is also equal to \( \mathbf{b}^T \mathbf{a} \). Two vectors that have a zero inner product are said to be orthogonal.

The norm of a vector \( \mathbf{a} \) is

\[ \| \mathbf{a} \| = \sqrt{\mathbf{a}^T \mathbf{a}}, \]

obviously a nonnegative number. A unit vector is a vector with norm one.

#### Examples

The vector \( \mathbf{a} = (2, -1, 0) \) corresponds to row vector \( \mathbf{a}^T = [2, -1, 0] \) and to column vector

\[ \mathbf{a} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix}. \]
The inner product of $a$ and $b = (1, 0, -1)$ is

$$a^T b = [2, -1, 0] \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = 2 \cdot 1 + (-1) \cdot 0 + 0 \cdot (-1) = 2 ,$$

and the norm of $a$ is

$$\|a\| = \sqrt{a^T a} = \sqrt{2 \cdot 2 + (-1) \cdot (-1) + 0 \cdot 0} = \sqrt{5} \approx 2.2361 .$$

The vector

$$c = \frac{1}{\sqrt{5}} a = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.8944 \\ -0.4472 \\ 0 \end{bmatrix}$$

has unit norm:

$$\|c\| = \sqrt{c^T c} = \sqrt{\left(\frac{2}{\sqrt{5}}\right)^2 + \left(\frac{-1}{\sqrt{5}}\right)^2 + \left(\frac{0}{\sqrt{5}}\right)^2} = \sqrt{\frac{2^2 + (-1)^2 + 0^2}{5}} = \sqrt{\frac{5}{5}} = \sqrt{1} = 1 .$$

### 3 Linear, Deterministic, Stationary, Discrete Dynamic Systems

The book shows examples of scalar, first order, dynamic systems of the form

$$a(n + 1) = ca(n) ,$$

possibly with a constant input,

$$a(n + 1) = ca(n) + u$$

as well as systems of dynamic systems of the form

$$a(n + 1) = Aa(n) + Bb(n)$$

$$b(n + 1) = Ca(n) + Db(n) .$$

In recitation, you saw scalar, second order, dynamic systems of the form

$$a(n + 1) = 2ba(n) - ca(n - 1)$$

and saw plots of their evolution.

These three types of systems provide a rather rich repertoire of responses, and are all instantiations of deterministic, linear, stationary, discrete dynamic systems (without “scalar”). To keep the mathematics simple, we limit ourselves to $2 \times 2$ systems, in which only at most two variables $a(n)$ and $b(n)$ appear.

As we saw in the introduction, a dynamic system is a system with inputs and outputs that vary over time. A model of the system is a mathematical description of how the quantities of interest change with time. The following are simple examples of a dynamic system:

- An electric circuit, whose input is the current in a given branch and whose output is a voltage across a pair of nodes.
• A chemical reactor, whose inputs are the external temperature, the temperature of the gas being supplied, and the supply rate of the gas. The output can be the temperature of the reaction product.

• A mass suspended from a spring. The input is the force applied to the mass and the output is the position of the mass.

In all these examples, what is input and what is output is a choice that depends on the application. Also, all the quantities in the examples vary continuously with time. In other cases, as for instance for switching networks and computers, it is more natural to consider time as a discrete variable. If time varies continuously, the system is said to be continuous; if time varies discretely, the system is said to be discrete. Even if the system per se is continuous, it is possible to restrict its description to a discrete set of points in time, so a discrete model can describe a continuous system. This Section considers only discrete models of dynamic systems.

Given a dynamic system, continuous or discrete, the modeling problem is to somehow correlate inputs (causes) with outputs (effects). The examples above suggest that the output at time \( t \) cannot be determined in general by the value assumed by the input quantity at the same point in time. Rather, the output is the result of the entire history of the system. An effort of abstraction is therefore required, which leads to postulating a new quantity, called the state, which summarizes information about the past and the present of the system. Specifically, the value \( x(t) \) taken by the state at time \( t \) must be sufficient to determine the output at the same point in time. Also, knowledge of both \( x(t_1) \) and \( u([t_1, t_2)) \), that is, of the state at time \( t_1 \) and the input over the interval \( t_1 \leq t < t_2 \), must allow computing the state (and hence the output) at time \( t_2 \). For the mass attached to a spring, for instance, the state could be the position and velocity of the mass. In fact, the laws of classical mechanics allow computing the new position and velocity of the mass at time \( t_2 \) given its position and velocity at time \( t_1 \) and the forces applied over the interval \( [t_1, t_2) \). Furthermore, in this example, the output \( y \) of the system happens to coincide with one of the two state variables, and is therefore always deducible from the latter.

Thus, in a dynamic system the input affects the state, and the output is a function of the state. For a discrete system, the way that the input changes the state at time instant number \( n \) into the new state at time instant \( n + 1 \) can be represented by a simple equation:

\[
x(n + 1) = f(x(n), u(n), n)
\]

where \( f \) is some function that represents the change, and \( u(n) \) is the input at time \( n \). Similarly, the relation between state and output can be expressed by another function:

\[
y(n) = h(x(n), n) .
\]

A discrete dynamic system is completely described by these two equations and an initial state \( x(0) = x_0 \). In general, the quantities \( x, u, y \) are vectors.

A discrete, dynamic system is linear if both \( f \) and \( h \) are linear. If this is the case, they can be
described by matrix-vector multiplications as follows:

\[
\begin{align*}
    x(0) &= x_0 \\
    x(n+1) &= F(n)x(n) + G(n)u(n) \\
    y(n) &= H(n)x(n) .
\end{align*}
\]

The notation \( F(n) \) means that for every point in time \( n \) there is in general a different matrix \( F \), and similarly for \( G(n) \) and \( H(n) \). A linear, discrete, dynamic system is \textit{stationary} if the matrices \( F(n), G(n), H(n) \) that describe the two functions \( f \) and \( h \) do not depend on the time variable \( n \):

\[
F(n) = F , \quad G(n) = G , \quad H(n) = H ,
\]

so that the model becomes

\[
\begin{align*}
    x(0) &= x_0 \\
    x(n+1) &= Fx(n) + Gu(n) \\
    y(n) &= Hx(n) .
\end{align*}
\] (1)

If the state \( x \) has \( k \) components, the input \( u \) has \( m \), and the output \( y \) has \( p \), then the matrices \( F, G, H \) have the following sizes:

\[
F \text{ is } k \times k , \quad G \text{ is } k \times m , \quad H \text{ is } p \times k .
\]

**Examples**

We now show that all the dynamic systems we have seen so far are linear, deterministic, stationary, discrete dynamic systems, that is, they have the form (1).

The first order system

\[
a(n+1) = ca(n) + u
\]

can be put in the form (1) with

\[
x(n) = a(n) , \quad F = c , \quad G = 1 , \quad u(n) = u , \quad H = 1
\]

so all matrices and vectors are \( 1 \times 1 \). If the input is absent, we set \( G = 0 \). The \( 2 \times 2 \) system

\[
\begin{align*}
    a(n+1) &= Aa(n) + Bb(n) \\
    b(n+1) &= Ca(n) + Db(n)
\end{align*}
\]

has

\[
x(n) = \begin{bmatrix} a(n) \\ b(n) \end{bmatrix} , \quad F = \begin{bmatrix} A & B \\ C & D \end{bmatrix} , \quad G = 0 , \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
\]

assuming that both quantities \( a(n) \) and \( b(n) \) are of interest, and are therefore listed as outputs. If only \( a(n) \) were of interest, we would set

\[
H = \begin{bmatrix} 1 & 0 \\ \end{bmatrix}
\]
The scalar, second order, dynamic systems of the form

\[ a(n + 1) = 2ba(n) - ca(n - 1) \]  

requires just a bit more work. We let

\[
\begin{bmatrix}
  x_1(n) \\
  x_2(n)
\end{bmatrix}
= \begin{bmatrix}
  a(n) \\
  a(n - 1)
\end{bmatrix}
\]

so that

\[
\begin{bmatrix}
  x_1(n + 1) \\
  x_2(n + 1)
\end{bmatrix}
= \begin{bmatrix}
  2bx_1(n) - cx_2(n) \\
  x_1(n)
\end{bmatrix}
= \begin{bmatrix}
  2b & -c \\
  1 & 0
\end{bmatrix}
\begin{bmatrix}
  x_1(n) \\
  x_2(n)
\end{bmatrix}
= \begin{bmatrix}
  2b & -c \\
  1 & 0
\end{bmatrix}
\begin{bmatrix}
  a(n) \\
  a(n - 1)
\end{bmatrix}
\]

This has the form

\[ x(n + 1) = Fx(n) \quad \text{with} \quad F = \begin{bmatrix}
  2b & -c \\
  1 & 0
\end{bmatrix}. \]

Note that the second row of \( F \) merely copies the first component of \( x(n) \) (that, is \( a(n) \)) into the second of \( x(n + 1) \). In other words, the value \( a(n) \) is the “current” value at time \( n \), and becomes the “most recent past” value at time \( n + 1 \). Because the right-hand side of equation (2) relies on two past values of \( a \), the state \( x(n) \) of the system has two entries, as two values need to be stored at all times to allow the recursion (2) to be computed. Of these two values, on the other hand, only the most recent (that is, the first component \( a(n) \) of \( x(n) \)) is of interest as an output to the system, so that

\[ H = \begin{bmatrix}
  1 & 0
\end{bmatrix}. \]

In summary, the recursion (2) is of the form (1) with

\[
\begin{bmatrix}
  x_1(n + 1) \\
  x_2(n + 1)
\end{bmatrix}
= \begin{bmatrix}
  2b & -c \\
  1 & 0
\end{bmatrix}
\begin{bmatrix}
  x_1(n) \\
  x_2(n)
\end{bmatrix}, \quad G = 0, \quad H = \begin{bmatrix}
  1 & 0
\end{bmatrix}. \]

So if we understand Linear, Deterministic, Stationary, Discrete Dynamic Systems (LDSDDS), we understand at least all the examples above in a unified way.

Such a system can be understood most easily by first understanding the evolution of its state \( x(n) \). If we know that, then the output \( y(n) \) can be computed through a simple transformation of \( x(n) \) through the output matrix \( H \). The evolution of the state, on the other hand, can be described as the sum of two terms, each of which is most easily understood by itself. The first term is the \textit{free evolution} \( x_f(n) \), which is the sequence of state vectors that would be obtained in the absence of input. The second term if the \textit{forced response} \( x_r(n) \), which accounts for the input but assumes, conversely, that the initial state \( x_0 \) is zero. Thus, \( x_f(n) \) is the solution to the recurrence

\[
\begin{align*}
  x(0) &= x_0, \\
  x(n + 1) &= Fx(n),
\end{align*}
\]

\[ (3) \text{ and } (4). \]
while \( x_r(n) \) is the solution to the recurrence
\[
\begin{align*}
x(0) & = 0 \\
x(n + 1) & = Fx(n) + Gu(n) .
\end{align*}
\]
Because of the linearity of the system, it is easy to show by direct substitution that the sum of these two terms,
\[x(n) = x_f(n) + x_r(n),\]
solves the complete system
\[
\begin{align*}
x(0) & = x_0 \\
x(n + 1) & = Fx(n) + Gu(n) .
\end{align*}
\]

Here is the proof: Since the free evolution \( x_f(n) \) satisfies equation (3), we must have \( x_f(0) = x_0 \). Since the forced response satisfies equation (5), we must have \( x_r(0) = 0 \). So
\[x(0) = x_f(0) + x_r(0) = x_0 + 0 = x_0 ,\]
so \( x(n) \) satisfies the initialization equation (7) of the complete system.

Similarly, since the free evolution \( x_f(n) \) satisfies equation (4), we must have \( x_f(n + 1) = Fx_f(n) \). Since the forced response satisfies equation (6), we must have \( x_r(n + 1) = Fx_r(n) + Gu(n) \). So
\[
\begin{align*}
x(n + 1) & = x_f(n + 1) + x_r(n + 1) = Fx_f(n) + Fx_r(n) + Gu(n) \\
& = F[x_f(n) + x_r(n)] + Gu(n) = Fx(n) + Gu(n) ,
\end{align*}
\]
which shows that \( x(n) \) satisfies the recurrence equation (8) of the complete system. 

The free evolution and the forced response can be found by unrolling the respective recursions. For the free evolution, equations (3) and (4) yield the following:
\[
\begin{align*}
x_f(0) & = x_0 \\
x_f(1) & = Fx_f(0) = Fx_0 \\
x_f(2) & = Fx_f(1) = F^2x_0
\end{align*}
\]
where \( F^2 = FF \) is the product of the matrix \( F \) with itself. The pattern is obvious:
\[x_f(n) = F^n x_0 ,\]
and can be verified by direct substitution into equations (3) and (4).

For the forced response, equations (5) and (6) yield the following:
\[
\begin{align*}
x_r(0) & = 0 \\
x_r(1) & = Fx_r(0) + Gu(0) = Gu(0) \\
x_r(2) & = Fx_r(1) + Gu(1) = FGu(0) + Gu(1) \\
x_r(3) & = Fx_f(2) + Gu(2) = F^2 Gu(0) + FGu(1) + Gu(2) .
\end{align*}
\]
Again, the pattern is clear:
\[ x_r(n) = \sum_{i=0}^{n-1} F_i G u(n - i - 1), \]
and can be verified by direct substitution into equations (5) and (6).

Thus, the complete system (7, 8) has solution
\[ x(n) = x_f(n) + x_r(n) = F^n x_0 + \sum_{i=0}^{n-1} F_i G u(n - i - 1). \]

Finally, since the output \( y(n) \) of the original system is
\[ y(n) = H x(n), \]
we also have the output of the full system. Table 1 summarizes these results.

**State Space**

The three equations in (9) can be interpreted geometrically. For simplicity, let us focus on the state (as opposed to the output \( y \)), and assume that there is no input \( (G = 0) \). Then the dynamic system simplifies to the following recursion for the state \( x(n) \):
\[ x(0) = x_0 \]
\[ x(n + 1) = F x(n) \]

Since \( x_0 \) has two entries, we can thin of these as the Cartesian coordinates on a plane, called the state space. So the initial state \( x_0 \) can be drawn as a point in state space. The second equation above, when applied to \( x(0) \), becomes
\[ x(1) = F x(0) \]
and yields a new point \( x(1) \). Because of this, the matrix \( F \) can be seen as a transformation from \( x(0) \) to \( x(1) \), and this transformation has the same mathematical expression (namely, \( F \)) regardless of the value \( x_0 \) assumed by the initial state. Moreover, the state equation
\[ x(n + 1) = F x(n) \]
holds for any state \( x(n) \), and \( F \) is still the relevant transformation.

In other words, we can interpret this recursion as taking a point \( x(0) = x_0 \) in state space, transforming it to a new point \( x(1) \), and then transforming that point to a new point \( x(2) \) in turn, and so forth. Figure 2 shows two examples that repeat this transformation for different matrices \( F \).

If the free evolution of interest is the first component \( x_1(n) \) of the state vector \( x(n) \), then the output matrix is
\[ H = \begin{bmatrix} 1, & 0 \end{bmatrix} \]
A $k \times k$ linear, deterministic, stationary, discrete dynamic system is represented by a model of the form

\[
\begin{align*}
    x(0) &= x_0 \\
    x(n+1) &= Fx(n) + Gu(n) \\
    y(n) &= Hx(n).
\end{align*}
\]  

(9)

The $k \times 1$ column vector $x(n)$ is the state of the system, the $m \times 1$ column vector $u(n)$ is the input, and the $p \times 1$ column vector $y(n)$ is the output. The known vector $x_0$ is the initial state, and the system matrix $F$, input matrix $G$, and output matrix $H$ have dimensions $k \times k$, $k \times m$, and $p \times k$, respectively.

The free evolution of the system (9) is the solution to the recurrence

\[
\begin{align*}
    x(0) &= x_0 \\
    x(n+1) &= Fx(n),
\end{align*}
\]

and is equal to

\[
x_f(n) = F^n x_0.
\]

The forced evolution of the system (9) is the solution to the recurrence

\[
\begin{align*}
    x(0) &= 0 \\
    x(n+1) &= Fx(n) + Gu(n),
\end{align*}
\]

and is equal to

\[
x_r(n) = \sum_{i=0}^{n-1} F^i Gu(n - i - 1).
\]

The evolution of the system (9) is the sum

\[
x(n) = x_f(n) + x_r(n) = F^n x_0 + \sum_{i=0}^{n-1} F^i Gu(n - i - 1),
\]

(10)

and the output of the system (9) is

\[
y(n) = Hx(n) = HF^n x_0 + H \sum_{i=0}^{n-1} F^i Gu(n - i - 1).
\]

(11)

Table 1: The Linear, Discrete, Stationary, Discrete Dynamic System (LDSDDS).
Figure 2: Trajectories of the state vector $\mathbf{x}(n)$ starting from the point $\mathbf{x}(0) = \mathbf{x}_0 = [0.3, 0]^T$ for four different matrices $F$. 
and the output equation reads

\[ y(n) = Hx(n) = \begin{bmatrix} 1, & 0 \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} = x_1(n). \]

Geometrically, the output \( y(n) \) is thus the projection of the state point onto the \( x_1 \) axis. The free evolutions for the systems in Figure 2 are shown in Figure 3.

For completeness, Table 2 gives, without proof, a closed form expression for the power \( F^n \) of a \( 2 \times 2 \) matrix, which appears in the state evolution recursion (10) and the output evolution recursion (11) for the LDSDDS described by equations (9).

In the expression (13) for \( F^n \), the two (possibly complex) diagonal terms \( \lambda_1^n, \lambda_2^n \) are exponentials in \( n \), and decay or grow exponentially, depending on the magnitudes of the eigenvalues \( \lambda_1 \) and \( \lambda_2 \). If the eigenvalues are complex, then they are each other’s complex-conjugates, because they are the solutions of a second order equation with real coefficients, the characteristic equation (12).

The term \( t(n) \) in expression (13) is a bit more complicated, but its behavior as \( n \to \infty \) is still determined by the magnitudes of the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) of \( F \), in the same way that the two diagonal terms \( \lambda_1^n, \lambda_2^n \) are. Thus, since the two matrices \( U \) and \( U^* \) in equation (13) are fixed, the eventual fate of \( F^n \) as \( n \to \infty \) is determined by the eigenvalues \( \lambda_1 \) and \( \lambda_2 \) of \( F \) as follows:

<table>
<thead>
<tr>
<th>( F )</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
1.71 & -0.707 \\
1 & 0
\end{bmatrix}
\] | \( \lambda_1 = 1.01 > 1 \) , \( \lambda_2 = 0.7 < 1 \) |
| \[
\begin{bmatrix}
1.80 & -0.810 \\
1 & 0
\end{bmatrix}
\] | \( \lambda_1 = 0.9 < 1 \) , \( \lambda_2 = 0.9 < 1 \) |
| \[
\begin{bmatrix}
1.24 & -0.970 \\
1 & 0
\end{bmatrix}
\] | \( \lambda_{1,2} = 0.62 \pm 0.7652i \) , \( |\lambda_{1,2}| = 0.9849 < 1 \) |
| \[
\begin{bmatrix}
1.24 & -1.030 \\
1 & 0
\end{bmatrix}
\] | \( \lambda_{1,2} = 0.62 \pm 0.8035i \) , \( |\lambda_{1,2}| = 1.0149 > 1 \) |

These different cases are evident in the trajectories and free evolutions shown in Figures 2 and 3. The eigenvalues for these four cases are as follows:
Figure 3: Free evolutions \( y(n) = x_1(n) \) starting from the state \( x(0) = x_0 = [0.3, 0]^T \) for the four matrices \( F \) in Figure 2.
The powers $F^n$ for $n = 0, 1, \ldots$ of any $2 \times 2$ matrix $F = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ with real entries can be computed in closed form as follows:

- Let $\lambda_1$ and $\lambda_2$ be the two (possibly complex, possibly coincident) solutions (in either order) of the following characteristic equation for $F$:
  \begin{equation}
  \lambda^2 - (a + d)\lambda + (ad - bc) = 0
  \end{equation}
  ($\lambda_{1,2}$ are the eigenvalues of $F$, $a + d$ is its trace, and $ad - bc$ is its determinant).
- Let $u_1 = [u_1, u_2]^T$ be a (possibly complex) unit-norm vector that satisfies the system $F_1 u = 0$ where $F_1 u = \begin{bmatrix} a - \lambda_1 & b \\ c & d - \lambda_1 \end{bmatrix}$.

Specifically, if $F_1$ is all zeros, let $u_1 = [1, 0]^T$. Otherwise, let $[f_1, f_2]$ be a nonzero row of $F_1$ (it does not matter which), and let $u_1 = [-f_2^*, f_1^*] / \sqrt{f_1^2 + f_2^2}$. Here, $x^*$ denotes the complex-conjugate of $x$ (If $x = r + ic$ is a complex number with real part $r$ and complex part $c$, then the complex-conjugate of $x$ is $x^* = r - ic$).

- Define the (possibly complex) matrices
  \[
  U = \begin{bmatrix} u_1 & -u_2^* \\ u_2 & u_1^* \end{bmatrix}, \quad U^* = \begin{bmatrix} u_1^* & u_2^* \\ -u_2 & u_1 \end{bmatrix} \quad \text{and} \quad T = \begin{bmatrix} \lambda_1 & \tau \\ 0 & \lambda_2 \end{bmatrix} = U^* F U
  \]
  ($T$ is provably triangular, that is, the first entry of its second row is zero, and its diagonal terms are the eigenvalues of $F$.)
- Then, for all $n = 0, 1, 2, \ldots$, the following expression holds:
  \[
  F^n = U \begin{bmatrix} \lambda_1^n & t(n) \\ 0 & \lambda_2^n \end{bmatrix} U^*
  \]
  where
  \[
  t(n) = \begin{cases} 
  \frac{\lambda_1^n - \lambda_2^n}{\lambda_1 - \lambda_2} & \text{if } \lambda_1 \text{ and } \lambda_2 \text{ are real and distinct} \\
  \tau n^{\lambda_1 - \lambda_2} & \text{if } \lambda_1 = \lambda_2 = \lambda \text{ (and therefore real)} \\
  \tau \rho^{n-1} \frac{\sin n\omega}{\sin \omega} & \text{if } \lambda_{1,2} = \rho e^{\pm i\omega} \text{ are complex-conjugate, with } \omega > 0
  \end{cases}
  \]

Table 2: Closed-form expressions for the state and output evolution of a $2 \times 2$ LDSDDS.
3 LINEAR, DETERMINISTIC, STATIONARY, DISCRETE DYNAMIC SYSTEMS