Appendix E

Linear Algebra Review

In this review we consider linear equations of the form

$$Ax = b, \text{ where } x \in \mathbb{R}^n, \ b \in \mathbb{R}^m, \text{ and } A \in \mathbb{R}^{m \times n}.$$

Such equations arise often in this textbook.

This review is a succinct sumary of some of the properties used in this textbook. This review does not provide derivations nor give pencil-and-paper solution methods as often found in linear algebra textbooks (for that, please consult a more comprehensive reference, such as [1, 2, 3], or the relevant Wikipedia pages). Instead, it is assumed you have software (e.g., MATLAB) that can do calculations for you.

Important! In this review, all elements of the matrix A are real, as is the case for all matrices you find in this book. Some results below do not generalize to matrices whose elements are complex.

An $m \times n$ matrix A has m rows and n columns, written

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = [a_1 \ a_2 \ \cdots \ a_n],$$

where the columns of A are written as the column vectors $a_1, \ldots, a_n$. The

transpose of A exchanges the rows and columns and is written

$$A^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} a_1^{\mathrm{T}} \\ a_2^{\mathrm{T}} \\ \vdots \\ a_n^{\mathrm{T}} \end{bmatrix}.$$

A complex number s is written $s = a + bj$, and the operators $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ extract the real and imaginary portions of a complex number, respectively. So $\mathrm{Re}(s) = a$ and $\mathrm{Im}(s) = b$. The magnitude of a complex number is $|s| = \sqrt{a^2 + b^2}$. The Euclidean norm of a real vector $x = (x_1, x_2, \ldots, x_n)$ is written $\|x\| = \sqrt{x_1^2 + \ldots + x_n^2}$.

E.1 Preliminaries

The matrix A can be viewed as a linear operator $A : \mathbb{R}^n \to \mathbb{R}^m$ mapping n-vectors (x) to m-vectors (b), i.e.,

$$Ax = b \text{ or, by elements, } \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

In the context of the mapping A, $\mathbb{R}^n$ is called the **domain** and $\mathbb{R}^m$ is called the **codomain**. The space of possible values Ax for all $x \in \mathbb{R}^n$ is called the **range**, **image**, or **column space** of the linear mapping A. The range is the same as the codomain if, for every $b \in \mathbb{R}^m$, there is an $x \in \mathbb{R}^n$ such that $Ax = b$. The range is sometimes called the **linear span** of the columns $\{a_1, \ldots, a_n\}$ of A, i.e., the set of all linear combinations of the columns, $k_1 a_1 + \cdots + k_n a_n$:

$$\mathrm{span}(\{a_1, \ldots, a_n\}) = \{k_1 a_1 + \cdots + k_n a_n \mid k_1, \ldots, k_n \in \mathbb{R}\}.$$

The matrix A deforms the space $\mathbb{R}^n$, perhaps by stretching, squeezing, shearing, rotating, reflecting, or even compressing to a lower-dimensional space (if the dimension of the range is less than the dimension of the domain). Examples of 2×2 matrices deforming one two-dimensional space to another are shown in Figure E.1.

The **rank** of a matrix, $\mathrm{rank}(A)$ (in MATLAB, `rank(A)`), is the dimension of its range, which can be no larger than the smaller of m and n. Equivalently, the rank is the number of **linearly independent** columns of A, i.e., columns
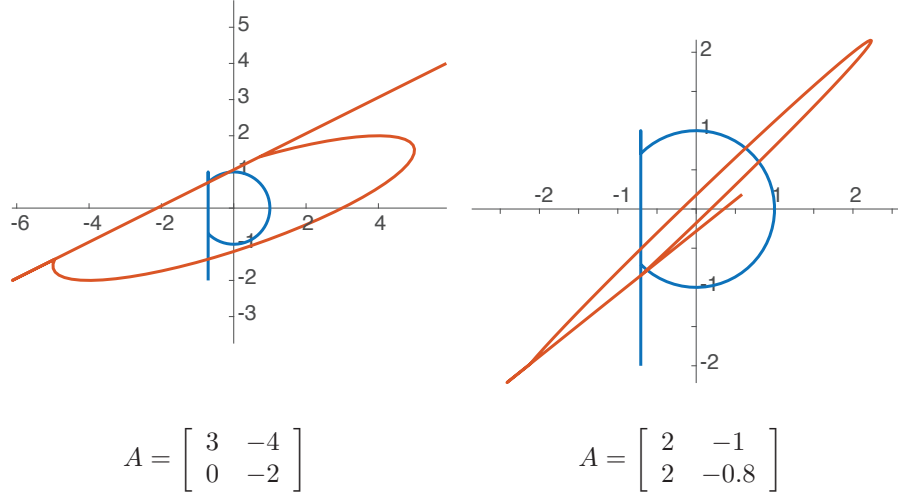
$$A = \begin{bmatrix} 3 & -4 \\ 0 & -2 \end{bmatrix} \qquad\qquad A = \begin{bmatrix} 2 & -1 \\ 2 & -0.8 \end{bmatrix}$$

Figure E.1: Two different A matrices mapping $\mathbb{R}^2$ to $\mathbb{R}^2$, as illustrated by the p transformed to Ap. The origin is a fixed point of the mapping.

which cannot be written as a linear combination of the other columns. (The number of linearly independent columns is the same as the number of linearly independent rows.) If the rank of the matrix is maximal (equal to the minimum of m and n), then we say the matrix is **full rank**. If A is not full rank, we often say it is **singular** or **rank deficient**. If the elements of A depend on some variable or variables θ (i.e., it can be written $A(\theta)$), and A is full rank for some θ and less than full rank for others, we say that $A(\theta)$ is singular at values of θ where $A(\theta)$ is less than full rank, and θ is called a **singularity** if $A(\theta)$ is singular.

The **null space** of a matrix, null(A) (in MATLAB, `null(A)`), also called the **kernel**, is the space of vectors x such that $Ax = 0$. The dimension of the null space is called the **nullity**, nullity(A). The **rank-nullity theorem** tells us that rank(A) + nullity(A) = n, the number of columns of A. As an example, a full rank 4×5 matrix has a rank of 4 and a nullity of 1, and the particular 4×5 matrix

$$A = \begin{bmatrix} 1 & 3 & -2 & 0 & 0 \\ -2 & -6 & 4 & 0 & 0 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

has rank$(A) = 2$, because three columns (e.g., a_2, a_3, and a_4) can be written as linear combinations of the other two (a_1 and a_5), e.g., $a_2 = 3a_1$, $a_3 = -2a_1 + 3a_5$, and since a_4 is the zero vector, it is trivially a linear combination of any of the columns with a coefficient of zero. Therefore the range of A is the space $k_1a_1 + k_5a_5$ for all $k_1, k_5 \in \mathbb{R}$. Since rank$(A) +$ nullity$(A) = n = 5$, the null space of A is three-dimensional. Using your favorite linear algebra software (e.g., `null(A)` in MATLAB), you would find that this three-dimensional linear subspace of $\mathbb{R}^5$ is spanned by the vectors $n_1 = [-0.95 \quad 0.32 \quad 0 \quad 0 \quad 0.01]^{\mathrm{T}}$, $n_2 = [0 \ 0 \ 0 \ 1 \ 0]^{\mathrm{T}}$, and $n_3 = [-0.06 \quad -0.19 \quad -0.31 \ 0 \ 0.93]^{\mathrm{T}}$. In other words, $Ax = 0$ for any x which can be expressed as $x = k_1n_1 + k_2n_2 + k_3n_3$, where each k_i is a real number.

E.2 Square Matrices

In this section we consider the special case where A is **square**: the number of rows equals the number of columns ($m = n$). In other words, both the domain and the codomain are $\mathbb{R}^n$.

A **diagonal matrix** is a square matrix with all elements not on the diagonal equal to zero, i.e., $a_{ij} = 0$ for $i \neq j$. An **identity matrix** I is a diagonal matrix with all elements along the diagonal equal to one. When it is helpful to specify that the identity matrix is $n \times n$, we can write I_n, e.g.,

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

As illustrated in Figure E.1, A maps $\mathbb{R}^n$ to $\mathbb{R}^n$ by stretching, squeezing, shearing, reflecting, rotating, etc. For some nonzero vectors v, the mapping may be particularly simple: Av is just a scaled version of v, i.e., $Av = \lambda v$. For example, if A is diagonal, then any vector aligned with an axis of the coordinate frame (e.g., a vector along the ith coordinate axis, $(0, \ldots, 0, x_i, 0, \ldots 0)$) is simply scaled by the corresponding element a_{ii} of the diagonal matrix.

For any λ and v satisfying $Av = \lambda v$, v is called an **eigenvector** of A and λ is the corresponding **eigenvalue** (in MATLAB, `[v,d]=eig(A)` returns the unit eigenvectors in `v` and the eigenvalues in `d`). In general, an eigenvalue λ is a complex scalar $a + bj \in \mathbb{C}$ and an eigenvector v is an element of $\mathbb{C}^n$, the space of n-vectors whose elements are complex numbers. In some cases the eigenvectors and eigenvalues are real, however ($\lambda \in \mathbb{R}, v \in \mathbb{R}^n$). If an eigenvector v is real, its corresponding eigenvalue λ must also be real, since it must satisfy $Av = \lambda v$.

An $n \times n$ matrix has n eigenvectors $v_1, \ldots, v_n$, each with an eigenvalue

$$A = \begin{bmatrix} 3 & -4 \\ 0 & -2 \end{bmatrix} \qquad\qquad A = \begin{bmatrix} 2 & -1 \\ 2 & -0.8 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \lambda_1 = 3 \qquad\qquad v_1 = \begin{bmatrix} 0.57 + 0.08j \\ 0.82 \end{bmatrix}, \lambda_1 = 0.6 + 0.2j$$

$$v_2 = \begin{bmatrix} 0.62 \\ 0.78 \end{bmatrix}, \lambda_2 = -2 \qquad\qquad v_2 = \begin{bmatrix} 0.57 - 0.08j \\ 0.82 \end{bmatrix}, \lambda_2 = 0.6 - 0.2j$$

$$\det(A) = \lambda_1\lambda_2 = -6 \qquad\qquad \det(A) = \lambda_1\lambda_2 = 0.4$$
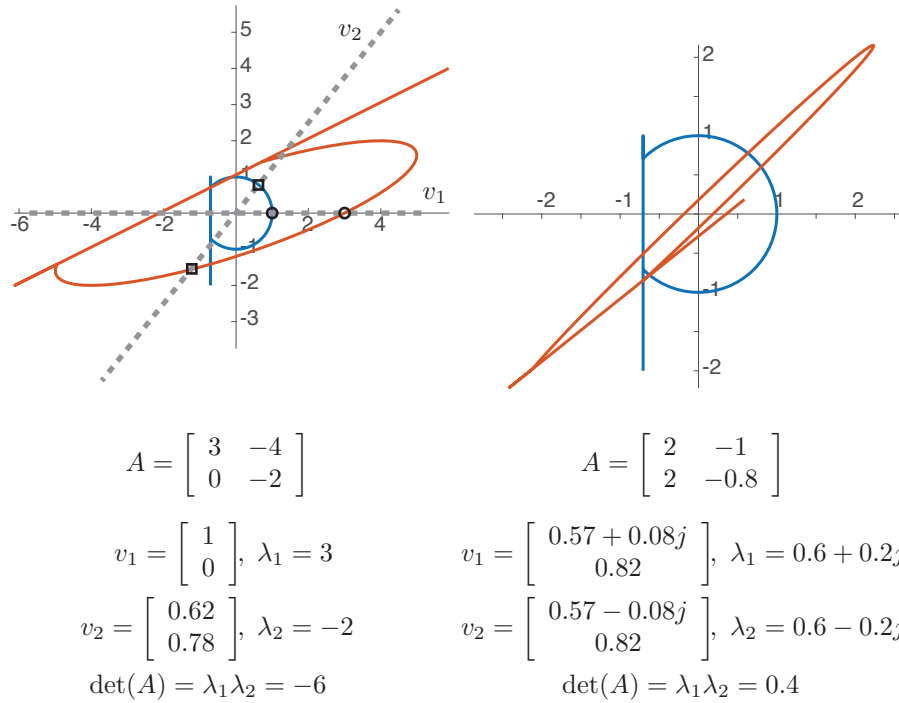
Figure E.2: Left: This A matrix has real eigenvalues and eigenvectors, and the one-dimensional spaces spanned by the individual eigenvectors are shown as dotted lines. The A matrix takes any vector along these lines and simply scales it (keeping it on the same line) by the corresponding eigenvalues, here 3 and -2. Two points on the original 'p,' shown as a square and a circle, are shown mapped to their new locations by the matrix A. Right: The eigenvalues and eigenvectors for this A matrix are not real, so there is no real vector $v \in \mathbb{R}^2$ and real value λ such that $Av = \lambda v$. There are no axes in the plane along which points are simply scaled.

$\lambda_1, \ldots, \lambda_n$. Complex eigenvalues always occur in complex conjugate pairs, e.g., $\lambda_1 = a + bj$ and $\lambda_2 = a - bj$.

When all eigenvectors and eigenvalues of a square matrix are real ($v_i \in \mathbb{R}^n, \lambda_i \in \mathbb{R}$), the eigendirections can be visualized in $\mathbb{R}^n$. Figure E.2 illustrates one A matrix where all eigenvectors and eigenvalues are real and another A matrix for which this is not the case.

Another simple example of a matrix without real eigenvectors and eigenval-

ues is

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \ \ v_1 = \begin{bmatrix} 0.71 \\ -0.71j \end{bmatrix}, \lambda_1 = j, \ \ v_2 = \begin{bmatrix} 0.71 \\ 0.71j \end{bmatrix}, \lambda_2 = -j.$$

This matrix rotates points in the plane by $\pi/2$ in the counterclockwise direction, and therefore there is no real vector $x \in \mathbb{R}^2$ for which Ax is just a scaled version of x.

The magnitudes of the eigenvalues $|\lambda_i|$ determine how much the space is stretched or contracted in the corresponding eigendirections. A magnitude greater than one ($|\lambda_i| > 1$) indicates that vectors are stretched in the corresponding eigendirection, and a magnitude less than one ($|\lambda_i| < 1$) indicates that vectors are contracted in the corresponding eigendirection. Eigenvalues play a key role in understanding the ability of feedback control laws to shrink tracking errors to zero (see Section E.7). A rotation matrix, like that in Equation (E.2), has unit magnitude eigenvalues: vectors are neither stretched nor contracted, simply rotated.

If each of the n eigenvalues is nonzero, then the matrix A is full rank, and the dimension of the range is the same as the dimension of the domain (n). If $p > 0$ of the n eigenvalues are zero, then the A matrix is not full rank (i.e., singular), and the dimension of the range of A is $n - p$.

The **determinant** of a square matrix A, $\det(A)$ (in MATLAB, det(A)), is a polynomial function of the elements of A. (Consult any standard reference for details.) The determinant satisfies

$$\det(A) = \lambda_1 \lambda_2 \cdots \lambda_n.$$

Thus A is full rank if and only if $\det(A) \neq 0$. If the determinant is zero, at least one of the eigenvalues is zero, and the matrix is not full rank. Since each eigenvalue expresses the scaling of vectors in one of the eigendirections, the determinant expresses whether the mapping A expands or contracts a volume in the space $\mathbb{R}^n$. In particular, a ball of volume V maps through A to an ellipsoid of volume $|\det(A)|V$.

The **characteristic polynomial** of a matrix A (in MATLAB, charpoly(A)) is $p(s) = \det(sI - A)$. The roots of this polynomial, e.g., the values of s satisfying $\det(sI - A) = 0$, are the eigenvalues of A. (To see this, consider that an eigenvalue λ of A must satisfy $Av = \lambda v$, or $(\lambda I - A)v = 0$, for a nonzero v, which can only be satisfied if $(\lambda I - A)$ is not full rank, i.e., $\det(\lambda I - A) = 0$.) The characteristic polynomial can be written in the forms

$$p(s) = \det(sI - A) = s^n + c_{n-1}s^{n-1} + \cdots + c_1 s + c_0 = (s - \lambda_1)(s - \lambda_2) \cdots (s - \lambda_n).$$

Another useful property of a square matrix A is its **trace**, $\text{tr}(A)$ (in MAT-LAB, `trace(A)`). The trace of a square matrix is simply the sum of the elements on its diagonal, which is equal to the sum of its eigenvalues:

$$\text{tr}(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i.$$

E.2.1   Inverse

The matrix **inverse** A^{-1} of A (in MATLAB, `inv(A)`) is the unique matrix that satisfies $AA^{-1} = A^{-1}A = I$. The matrix inverse exists only if A is full rank ($\det(A) \neq 0$). If the eigenvectors and eigenvalues of A are $\{v_i\}$ and $\{\lambda_i\}$, respectively, the eigenvectors and eigenvalues of A^{-1} are $\{v_i\}, \{1/\lambda_i\}$.

The matrix inverse can be used to solve $Ax = b$ for x, i.e., $x = A^{-1}b$, but there are more numerically accurate and computationally efficient methods for solving such equations. For example, MATLAB's algorithm for solving $Ax = b$ (the code is `x = A\b`) looks for faster and more accurate ways to solve the problem based on properties of A.

E.2.2   Pseudoinverse

If A is not full rank, then the inverse does not exist, but we can still calculate the Moore-Penrose **pseudoinverse** of A, denoted $A^{\dagger}$ (`pinv(A)` in MATLAB). The pseudoinverse has "inverse-like" properties and can be used to find solutions or approximate solutions to $Ax = b$, i.e., $x = A^{\dagger}b$. The pseudoinverse $A^{\dagger}$ is equivalent to the inverse A^{-1} when A is invertible. The pseudoinverse can also be calculated for non-square matrices, so the general description of the pseudoinverse is deferred to Section E.3.

E.2.3   Symmetric Matrices

A square matrix A is **symmetric** if it is equal to its transpose, $A = A^{\text{T}}$. A matrix A is **skew symmetric** if $A = -A^{\text{T}}$.

Every symmetric matrix A is **diagonalizable**, i.e., there exists an **orthogonal** matrix $Q \in \mathbb{R}^{n \times n}$ and a diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that

$$A = QDQ^{\text{T}}.$$

An orthogonal matrix Q is one satisfying $Q^{\text{T}}Q = QQ^{\text{T}} = I$, i.e., $Q^{-1} = Q^{\text{T}}$, and the columns of Q are unit vectors ($\|a_i\| = 1, i \in \{1, \ldots, n\}$) and orthogonal ($a_i^{\text{T}} a_j = 0, i \neq j$). Equivalently, the rows of Q are orthogonal unit vectors (also

called **orthonormal** vectors). The set of $n \times n$ orthogonal matrices is called the **orthogonal group** $O(n)$, and all orthogonal matrices have a determinant of $+1$ or -1. The set of orthogonal matrices with the determinant $+1$ is called the **special orthogonal group** $SO(n)$, also known as rotation matrices for $n = 2$ and $n = 3$. The matrix Q can always be chosen to be an element of $SO(n)$.

For a symmetric (and therefore diagonalizable) matrix $A = QDQ^{\mathrm{T}}$:

(a) Each diagonal element $d_i \in \mathbb{R}$ of D is an eigenvalue λ_i of A. Therefore all eigenvalues of symmetric matrices are real.

(b) The ith column of Q is the real eigenvector corresponding to the eigenvalue d_i; hence the eigenvectors of A are real and orthogonal. The columns of Q represent the axes of a new coordinate frame relative to the original coordinate frame in which the mapping A is represented. In this new coordinate frame the linear mapping by the matrix D is equivalent to the linear mapping represented by A in the original coordinate frame. In other words, if A is symmetric, there exists a rotated coordinate frame in which it would be diagonal.

A symmetric matrix A is **positive definite** if its eigenvalues are all positive. One consequence is the condition $x^{\mathrm{T}} A x > 0$ for all nonzero $x \in \mathbb{R}^n$. Positive-definite matrices have a number of important applications in robotics, for example in evaluating the manipulability of a robot (Chapter 5) and in representing the inertia of a rigid body or a robot (Chapter 8). Since positive-definite matrices are symmetric, they behave like diagonal matrices in the proper choice of coordinate frame. Figure E.3 illustrates two ellipsoid visualizations of a positive-definite matrix.

E.2.4   Square Matrices Representing Rigid-Body Motion

Square matrices are often used in the representation of rigid-body configurations and motion. These are discussed in more detail in Chapters 3 and 8; below is a brief summary.

A rigid-body orientation or rotation is expressed as a **rotation matrix** R in $SO(3)$, the group of 3×3 orthogonal matrices with determinant $+1$. Therefore the inverse of a rotation matrix R is R^{T}.

Rigid-body configurations and displacements are expressed as 4×4 matrices in the **special Euclidean group** $SE(3)$. A **transformation matrix** $T \in SE(3)$ can be written as

$$T = \left[\begin{array}{cc} R & p \\ 0 & 1 \end{array} \right],$$

$$A = \begin{bmatrix} 2 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 0.92 \\ 0.38 \end{bmatrix}, \ \lambda_1 = 2.21$$

$$v_2 = \begin{bmatrix} -0.38 \\ 0.92 \end{bmatrix}, \ \lambda_2 = 0.79$$

$$A = QDQ^{\mathrm{T}}, \ \text{where } Q = \begin{bmatrix} 0.92 & -0.38 \\ 0.38 & 0.92 \end{bmatrix} \text{ and } D = \begin{bmatrix} 2.21 & 0 \\ 0 & 0.79 \end{bmatrix}$$
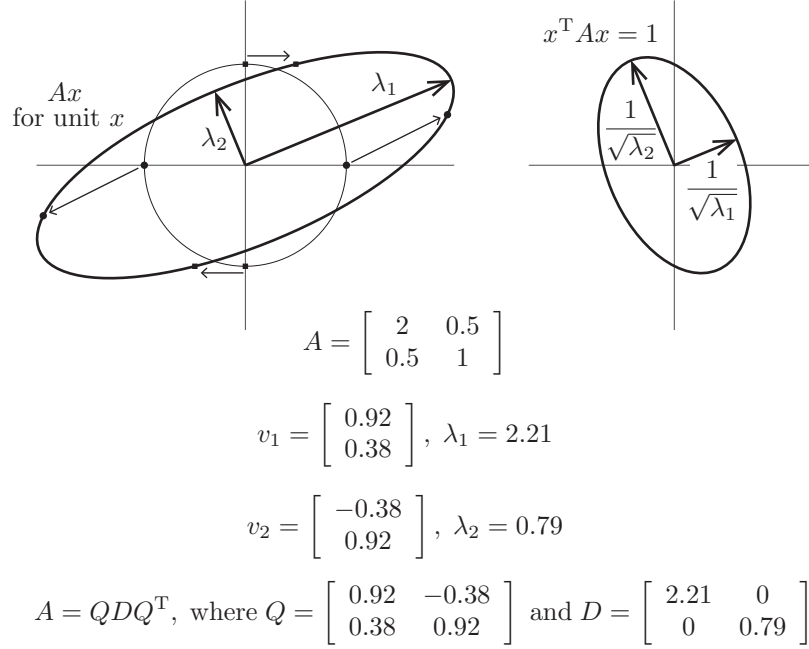
Figure E.3: Left: The positive-definite mapping A acting on the set of unit vectors x (including four specific vectors x) generates an ellipsoid, illustrating the positive real eigenvalues and the coordinate frame in which the mapping is diagonal. (The axes of this coordinate frame are the eigenvectors, which are the columns of Q.) Right: The set of vectors x on the ellipsoid $x^{\mathrm{T}}Ax = 1$. The principal semi-axes of the ellipsoid are aligned with the eigenvectors of A and have length $1/\sqrt{\lambda_i}$.

where $R \in SO(3)$, $p \in \mathbb{R}^3$, and the bottom row is $[0 \ 0 \ 0 \ 1]$. The column vector p represents a linear displacement and R is a rotation matrix. The inverse of a transformation matrix is

$$T^{-1} = \begin{bmatrix} R^{\mathrm{T}} & -R^{\mathrm{T}}p \\ 0 & 1 \end{bmatrix}.$$

The column vector $\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ can be expressed as the 3×3 skew-symmetric matrix

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix},$$

which satisfies $[\omega]p = \omega \times p$ for $p \in \mathbb{R}^3$. The set of all 3×3 skew symmetric matrices is denoted $so(3)$. The vector $\omega \in \mathbb{R}^3$ and its $so(3)$ representation $[\omega]$ are often used to represent the angular velocity of a rigid body.

The velocity of a rigid body is represented as a 6-vector twist, $\mathcal{V} = (\omega, v)$, where $\omega, v \in \mathbb{R}^3$. This 6-vector is sometimes represented in 4×4 matrix form as

$$[\mathcal{V}] = \left[\begin{array}{cc} [\omega] & v \\ 0 & 0 \end{array} \right] \in se(3).$$

Note that the notation $[\cdot]$ is used both to turn a 3-vector into an element of $so(3)$ and to turn a 6-vector into an element of $se(3)$.

The matrix $[\text{Ad}_{T_{ab}}] \in \mathbb{R}^{6 \times 6}$ can be used to change the representation of a 6-vector expressed in the frame {b} to a 6-vector expressed in the frame {a}, i.e., $\mathcal{V}_a = [\text{Ad}_{T_{ab}}]\mathcal{V}_b$, where

$$[\text{Ad}_T] = \left[\begin{array}{cc} R & 0 \\ {[p]}R & R \end{array} \right] \text{ if } T = \left[\begin{array}{cc} R & p \\ 0 & 1 \end{array} \right].$$

Given a twist $\mathcal{V}_1 = (\omega_1, v_1) \in \mathbb{R}^6$, the matrix

$$[\text{ad}_{\mathcal{V}_1}] = \left[\begin{array}{cc} [\omega_1] & 0 \\ {[v_1]} & [\omega_1] \end{array} \right] \in \mathbb{R}^{6 \times 6}$$

can be used to calculate the Lie bracket of the twists $\mathcal{V}_1$ and $\mathcal{V}_2$, $[\text{ad}_{\mathcal{V}_1}]\mathcal{V}_2$, as discussed in Chapter 8.

E.3   Non-square Matrices

A non-square matrix $A \in \mathbb{R}^{m \times n}$ is **tall** if it has more rows than columns ($m > n$) and **fat** (or **wide** or **broad**) if it has more columns than rows ($n > m$).

E.3.1   Pseudoinverse

A non-square matrix does not have an inverse, but if it is full rank, it can have either a **right inverse** or **left inverse**:

- If A is tall and full rank ($\text{rank}(A) = n$), then there exists a left inverse $A_{\text{L}}^{-1} \in \mathbb{R}^{n \times m}$ such that $A_{\text{L}}^{-1}A = I_n$.

- If A is fat and full rank ($\text{rank}(A) = m$), then there exists a right inverse $A_{\text{R}}^{-1} \in \mathbb{R}^{n \times m}$ such that $AA_{\text{R}}^{-1} = I_m$.

The left and right inverses are not unique, but examples are $A_{\mathrm{L}}^{-1} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ and $A_{\mathrm{R}}^{-1} = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$, respectively. You can verify easily that these are indeed left and right inverses.

These inverses are examples of the Moore-Penrose pseudoinverse $A^{\dagger}$. The pseudoinverse also exists for non-square matrices that are not full rank, as well as for all square matrices (full rank or singular). The pseudoinverse $A^{\dagger} \in \mathbb{R}^{n \times m}$ of any real matrix $A \in \mathbb{R}^{m \times n}$ satisfies the following conditions:

- $AA^{\dagger}A = A$

- $A^{\dagger}AA^{\dagger} = A^{\dagger}$

- $AA^{\dagger}$ is symmetric

- $A^{\dagger}A$ is symmetric

Also, $(A^{\dagger})^{\dagger} = A$, and the pseudoinverse $A^{\dagger}$ of an invertible square matrix A is equal to A^{-1}.

The pseudoinverse can be used to solve, or approximately solve, equations of the form $Ax = b$. If there is an entire space of solutions x to $Ax = b$, then $z = A^{\dagger}b$ satisfies $Az = b$ and $\|z\| \leq \|x\|$ for any x satisfying $Ax = b$. In other words, z minimizes the Euclidean norm among all solutions.

In the case that no x solves $Ax = b$, then $z = A^{\dagger}b$ satisfies the condition that $\|Az - b\| \leq \|Ax - b\|$ for all $x \in \mathbb{R}^n$. In other words, z comes as close to solving the equation as possible, in a least-squares sense.

E.4 Singular Value Decomposition

The pseudoinverse can be calculated using the **singular-value decomposition** (SVD). The singular-value decomposition of $A \in \mathbb{R}^{m \times n}$ expresses it as

$$A = U\Sigma V^{\mathrm{T}},$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ contains all zeros other than the nonnegative **singular values** $\sigma_{ii} \geq 0$ on the diagonal, which are more commonly written σ_i. For example, if A is a 4×5 matrix, then

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & 0 \end{bmatrix},$$

and if it is 5×4, then

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The columns of Σ are typically ordered so that $\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots$

In MATLAB, the command `[U,S,V] = svd(A)` returns the matrices U, Σ, and V. The pseudoinverse is then calculated as

$$A^\dagger = V\Sigma^\dagger U^{\mathrm{T}},$$

where $\Sigma^\dagger$ is the pseudoinverse of Σ, which is obtained by taking the transpose of Σ and then replacing every nonzero singular value σ_i by its reciprocal $1/\sigma_i$.

Another application of singular-value decomposition, relevant to matrix representations of orientation, is to project an arbitrary matrix A to the nearest orthogonal matrix (in a least-squares sense). For example, suppose you've performed a series of matrix multiplications of 3×3 rotation matrices, and the result at the end is A. Because of numerical roundoff errors, A may have drifted from the space of rotation matrices $SO(3)$, which you remember is a three-dimensional surface embedded in the nine-dimensional space defined by the elements of A. To find the orthogonal matrix R that is closest to A (in a least-squares sense), you could use the singular-value decomposition to express A as

$$A = U\Sigma V^{\mathrm{T}}.$$

If A is only slightly off of $SO(3)$ due to roundoff errors, the singular values σ_1, σ_2, and σ_3 should all be close to one, since all singular values of a rotation matrix are one. The closest orthogonal matrix R is obtained by setting each of the singular values *exactly* to one, i.e.,

$$R = UIV^{\mathrm{T}} = UV^{\mathrm{T}}.$$

Since U and V^{T} are both orthogonal (elements of $O(3)$), then $UV^{\mathrm{T}} \in O(3)$. If A is "close" to $SO(3)$, then R is exactly in $SO(3)$, the set of $O(3)$ matrices with determinant equal to $+1$. (If A is far from $SO(3)$, then $R = UV^{\mathrm{T}} \in O(3)$ may have a determinant of -1, meaning that it is not in $SO(3)$.)

The singular-value decomposition is useful in many other applications, so it is worth understanding it a bit. The columns of U are called the **left-singular vectors** of A and the columns of V are called the **right-singular vectors**. The singular values, left-singular vectors, and right-singular vectors satisfy the following conditions:

- The nonzero singular values σ_i are the square roots of the nonzero eigenvalues of the symmetric matrices $A^{\mathrm{T}}A$ and AA^{T}.

- The left-singular vectors of A are the orthonormal eigenvectors of the symmetric matrix AA^{T}.

- The right-singular vectors of A are the orthonormal eigenvectors of the symmetric matrix $A^{\mathrm{T}}A$.

The rank of A is equal to the number of nonzero singular values, p. The range of A is spanned by the p columns of U corresponding to the nonzero singular values (i.e., the first p columns of U, assuming the singular values are arranged in decreasing order in Σ). The nullity of A is $n - p$, and the null space of A is spanned by the last $n - p$ columns of V.

As an example, choose the 4×5 matrix

$$
A = \left[\begin{array}{ccccc}
1 & 3 & -2 & 0 & 0 \\
-2 & -6 & 4 & 0 & 0 \\
0 & 0 & 3 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{array}\right].
$$

As we saw at the end of Section E.1, this matrix has a rank of two and a nullity of three. The singular-value decomposition of A is

$$
U = \left[\begin{array}{cccc}
-0.44 & -0.09 & 0.89 & 0 \\
0.87 & 0.19 & 0.45 & 0 \\
0.21 & -0.98 & 0 & 0 \\
0 & 0 & 0 & 1
\end{array}\right], V = \left[\begin{array}{ccccc}
-0.26 & -0.17 & -0.95 & 0 & -0.06 \\
-0.77 & -0.52 & 0.32 & 0 & -0.19 \\
0.59 & -0.75 & 0 & 0 & -0.31 \\
0 & 0 & 0 & 1 & 0 \\
0.02 & -0.37 & 0.01 & 0 & 0.93
\end{array}\right],
$$

$$
\Sigma = \left[\begin{array}{ccccc}
8.54 & 0 & 0 & 0 & 0 \\
0 & 2.67 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{array}\right], \text{ i.e., } \sigma_1 = 8.54, \sigma_2 = 2.67, \sigma_3 = \sigma_4 = 0.
$$

There are $p = 2$ nonzero singular values, so $\mathrm{rank}(A) = 2$ and the range is spanned by the first two columns of A, $u_1 = [-0.44 \quad 0.87 \quad 0.21 \quad 0]^{\mathrm{T}}$ and $u_2 = [-0.09 \quad 0.19 \quad -0.98 \quad 0]^{\mathrm{T}}$. The nullity of A is $5 - p = 3$, and the null space is spanned by the last three columns of V, $v_3 = [-0.95 \; 0.32 \; 0 \; 0 \; 0.01]^{\mathrm{T}}$, $v_4 = [0 \; 0 \; 0 \; 1 \; 0]^{\mathrm{T}}$, and $v_5 = [-0.06 \quad -0.19 \quad -0.31 \; 0 \; 0.93]^{\mathrm{T}}$.

For $i \leq \min(m, n)$, the matrix $A \in \mathbb{R}^{m \times n}$ maps a vector aligned with the right-singular unit vector v_i (the ith column of V) to a vector aligned with the left-singular unit vector u_i (the ith column of U) with an amplification factor

of σ_i, i.e., $Av_i = \sigma_i u_i$. In the example 4×5 A matrix above, the unit vector v_1 maps through A to a vector $Av_1 = \sigma u_1$ with a magnitude $\|Av_1\| = \sigma_1 = 8.54$, and the unit vector v_2 maps through A to a vector $Av_2 = \sigma_2 u_2$ with a magnitude $\|Av_2\| = \sigma_2 = 2.67$. The singular values σ_3 and σ_4 are zero, so vectors aligned with v_3 and v_4 map through A to the zero vector.

The right- and left-singular vectors of a symmetric matrix A are equal, and equal to the eigenvectors of A. Furthermore, if A is also positive definite, its singular values are equal to its eigenvalues.

E.5   Ill-Conditioned Matrices

If a matrix A is full rank (all of its singular values σ_i are greater than zero), the **condition number** is defined as

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)},$$

the ratio of the largest and smallest singular values. If the condition number is large, the matrix is "nearly" singular and is called **ill conditioned** or **poorly conditioned**. Computations with ill-conditioned matrices may lead to outputs that are highly sensitive to small changes in the inputs.

For example, consider the ill-conditioned matrix

$$J = \left[\begin{array}{cc} 2 & 0 \\ 0 & 10^{-5} \end{array} \right],$$

which has singular values 2 and 10^{-5} and the condition number $\kappa(J) = 2 \times 10^5$. The exact pseudoinverse is

$$J^\dagger = \left[\begin{array}{cc} 0.5 & 0 \\ 0 & 10^5 \end{array} \right].$$

Now imagine that J is the Jacobian mapping a robot arm's joint speeds $\dot{\theta} = (\dot{\theta}_1, \dot{\theta}_2)$ to the velocity of its end-effector $v = (v_1, v_2)$, i.e., $v = J\dot{\theta}$. The near-singularity of J indicates that it is difficult to generate motion in the v_2 direction. If we calculate the commanded joint speeds $\dot{\theta}_{\text{com}}$ for a desired end-effector speed v_d using $\dot{\theta}_{\text{com}} = J^{-1} v_d = J^\dagger v_d$, then we will get a very large $|\dot{\theta}_2|$ if we ask (perhaps accidentally) for even a small velocity component v_2, due to the large matrix entry 10^5. This kind of large sensitivity in the output ($\dot{\theta}_{\text{com}}$) to small variations in the input (v_d) is undesirable.

One approach to dealing with ill conditioning is to treat sufficiently small singular values as zero. For example, in MATLAB, we can tell the pseudoinverse

function `pinv` to treat singular values below a particular tolerance as zero in the SVD calculation of the pseudoinverse. For the J matrix above, we can treat singular values below 10^{-4} as zero using the modified `pinv` command:

```
>> Jptol = pinv(J,1e-4)
 Jptol =
        0.5   0
          0   0
```

With this modified pseudoinverse `Jptol`, requested velocity components in the v_2 direction are ignored as unattainable instead of amplified unreasonably in the calculation of $\dot{\theta}_{\text{com}}$.

E.6 Some Matrix Identities

The matrix identities below are used in this book. You can find many more useful identities in most linear algebra textbooks, and *The Matrix Cookbook* [2] is an excellent free online reference.

$$(AB)C = A(BC) \qquad \text{associativity} \qquad (\text{E.1})$$

$$(A^{\text{T}})^{-1} = (A^{-1})^{\text{T}} \qquad \text{also written as } A^{-\text{T}} \qquad (\text{E.2})$$

$$(Ax)^{\text{T}} = x^{\text{T}}A^{\text{T}} \qquad (\text{E.3})$$

$$(AB)^{\text{T}} = B^{\text{T}}A^{\text{T}} \qquad (\text{E.4})$$

$$(ABC\ldots)^{\text{T}} = \ldots C^{\text{T}}B^{\text{T}}A^{\text{T}} \qquad (\text{E.5})$$

$$(AB)^{-1} = B^{-1}A^{-1} \qquad \text{if } A \text{ and } B \text{ are invertible} \qquad (\text{E.6})$$

$$(ABC\ldots)^{-1} = \ldots C^{-1}B^{-1}A^{-1} \qquad \text{if each matrix is invertible} \qquad (\text{E.7})$$

$$[a] = -[a]^{\text{T}} \qquad \text{for } a \in \mathbb{R}^3, \text{ where } [\cdot] \text{ denotes} \qquad (\text{E.8})$$
$$\text{the } so(3) \text{ representation}$$

$$[a]b = -[b]a \qquad \text{for } a, b \in \mathbb{R}^3 \qquad (\text{E.9})$$

$$[a][b] = ([b][a])^{\text{T}} \qquad \text{for } a, b \in \mathbb{R}^3 \qquad (\text{E.10})$$

$$R[\omega]R^{\text{T}} = [R\omega] \qquad \text{for } \omega \in \mathbb{R}^3, R \in SO(3) \qquad (\text{E.11})$$

E.7 Applications in Feedback Control

Because of the importance of linear algebra to control theory, and the importance of control theory to robotics, in this section we provide an example of how
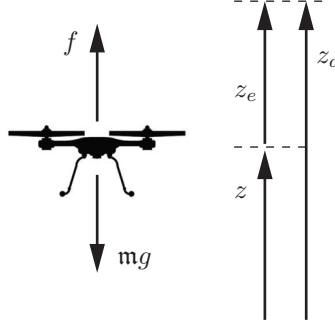
Figure E.4: The actual height of the quadrotor is z, the desired height is z_d, and the error is $z_e = z_d - z$. The upward control thrust is f and the downward force of gravity is $\mathfrak{m}g$.

some of the concepts above are used in control theory.

Suppose we are trying to control the height of a quadrotor above the ground (Figure E.4). The actual height is z, the desired height is z_d, and the error is $z_e = z_d - z$. The quadrotor has a force $\mathfrak{m}g$ pulling it downward and it is subject to a linear drag force opposing its upward/downward velocity, $-b\dot{z}$, where $b > 0$. Our control is the amount of upward thrust force f from the propellers.

With these definitions, the dynamics of the quadrotor are

$$\mathfrak{m}\ddot{z} + b\dot{z} = f - \mathfrak{m}g = u, \tag{E.12}$$

where u is the "pseudocontrol" $f - \mathfrak{m}g$. We will design a control law for u, then calculate the actual control input as $f = u + \mathfrak{m}g$.

A common type of feedback controller is a proportional-derivative (PD) controller

$$u = k_p z_e + k_d \dot{z}_e, \; k_p, k_d > 0,$$

where the term $k_p z_e$ acts like a virtual spring that tries to drive the error z_e to zero, and the term $k_d \dot{z}_e$ acts like a virtual damper that tries to drive the rate of change of error $\dot{z}_e$ to zero. Plugging this controller in for u in Equation (E.12), we get

$$\mathfrak{m}\ddot{z} + b\dot{z} = k_p z_e + k_d \dot{z}_e. \tag{E.13}$$

If we are trying to stabilize the height of the quadrotor to a constant z_d, then $\dot{z}_d = \ddot{z}_d = 0$. Plugging in $\dot{z} = \dot{z}_d - \dot{z}_e = -\dot{z}_e$ and $\ddot{z} = \ddot{z}_d - \ddot{z}_e = -\ddot{z}_e$, we can rewrite the controlled dynamics (E.13) as the controlled error dynamics

$$\mathfrak{m}\ddot{z}_e + (b + k_d)\dot{z}_e + k_p z_e = 0. \tag{E.14}$$

To write the second-order differential equation (E.14) as two first-order differential equations, we define

$$x_1 = z_e$$
$$x_2 = \dot{x}_1 = \dot{z}_e$$

and get the equations

$$\dot{x}_1 = x_2 \tag{E.15}$$

$$\dot{x}_2 = -\frac{k_p}{\mathfrak{m}} x_1 - \frac{b + k_d}{\mathfrak{m}} x_2, \tag{E.16}$$

where Equation (E.16) is just a restatement of the second-order controlled error dynamics (E.14). We can write Equations (E.15) and (E.16) as the vector equation

$$\dot{x} = Bx,$$

where

$$B = \left[\begin{array}{cc} 0 & 1 \\ -k_p/\mathfrak{m} & -(b + k_d)/\mathfrak{m} \end{array} \right].$$

If we choose $\mathfrak{m} = 1$ and $b = 0.1$ for the quadrotor, leaving the controls k_p and k_d as variables, then

$$B = \left[\begin{array}{cc} 0 & 1 \\ -k_p & -k_d - 0.1 \end{array} \right]. \tag{E.17}$$

The first-order vector differential equation $\dot{x} = Bx$ determines the evolution of the error in the height, $x_1(t) = z_e(t)$.

Often it is convenient to represent this error differential equation in discrete time. For example, the controller running on the quadrotor may read its sensors and evaluate its control law every Δt seconds. So, instead of representing the error dynamics as a continuous function of time $\dot{x}(t) = Bx(t)$, we could represent it at discrete times $c\Delta t$, $(c + 1)\Delta t$, $(c + 2)\Delta t$, etc. A simple discrete-time approximation to the continuous-time error dynamics $\dot{x}(t) = Bx(t)$ is

$$\frac{1}{\Delta t} \Big(x((c + 1)\Delta t) - x(c\Delta t) \Big) \approx Bx(c\Delta t).$$

Approximating $\approx$ by $=$, we get

$$x((c + 1)\Delta t) = Ax(c\Delta t), \text{ where } A = (I + B\Delta t).$$

Defining $k = c\Delta t$ and choosing $\Delta t = 0.1$, we get the discrete-time error dynamics

$$x(k + 1) = Ax(k), \text{ where } A = \left[\begin{array}{cc} 1 & 0.1 \\ -0.1k_p & 0.99 - 0.1k_d \end{array} \right]. \tag{E.18}$$

Now that we have our controlled quadrotor's error dynamics written in both discrete time as $x(k + 1) = Ax(k)$ and continuous time as $\dot{x} = Bx$, we can apply what we have learned about eigenvalues to understand the stability of the controlled system.

E.7.1   Discrete-Time Control

The error dynamics for a general discrete-time control system can be written

$$x(k+1) = Cx(k) + Du(k), \tag{E.19}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $C \in \mathbb{R}^{n \times n}$, and $D \in \mathbb{R}^{n \times m}$. Choosing a feedback controller $u = Kx$, where $K \in \mathbb{R}^{m \times n}$, the error dynamics become

$$x(k+1) = (C + DK)x(k) = Ax(k),$$

and for the quadrotor, A is given in Equation (E.18).

We say that the controlled error dynamics are **stable** if the error vector x converges to zero as we iteratively map it through A. The eigenvalues of A determine whether or not the error decays to zero. If the magnitudes of all the eigenvalues of A are less than one, then the iterated mapping contracts the error vector to zero, and the matrix A is called **convergent**. One major purpose of the controller $K \in \mathbb{R}^{m \times n}$ is to ensure that the eigenvalues λ_i of A satisfy $|\lambda_i| < 1$ for all i. The decay of the error to zero is faster for eigenvalue magnitudes closer to zero.

Figure E.5 shows examples of the quadrotor's error dynamics for different choices of k_p and k_d. Four different cases are shown: two stable cases ($|\lambda_1|, |\lambda_2| < 1$), one where both eigenvalues are real and one where the eigenvalues are complex conjugates; and two unstable cases, one where both eigenvalues are real with $|\lambda_1| > 1$ and $|\lambda_2| < 1$ (known as a "saddle," since error increases in one direction and decreases in the other, just as the height from the low point of a horse's saddle goes up in the forward-backward direction and down in the side-to-side direction) and one where the eigenvalues are complex conjugates with $|\lambda_1|, |\lambda_2| > 1$. Each case shows an initial circle of errors in the (x_1, x_2)-space and how it evolves under the iterated mapping by A. For the cases of the real eigenvectors and eigenvalues, most initial errors converge toward the line of the eigenvector with the larger eigenvalue magnitude. Initial errors exactly on the line of the eigenvalue with the smaller magnitude stay on that line throughout their evolution.

Stability depends on the particular control law. Another property of the control system (E.19), **controllability**, is independent of the control law. We
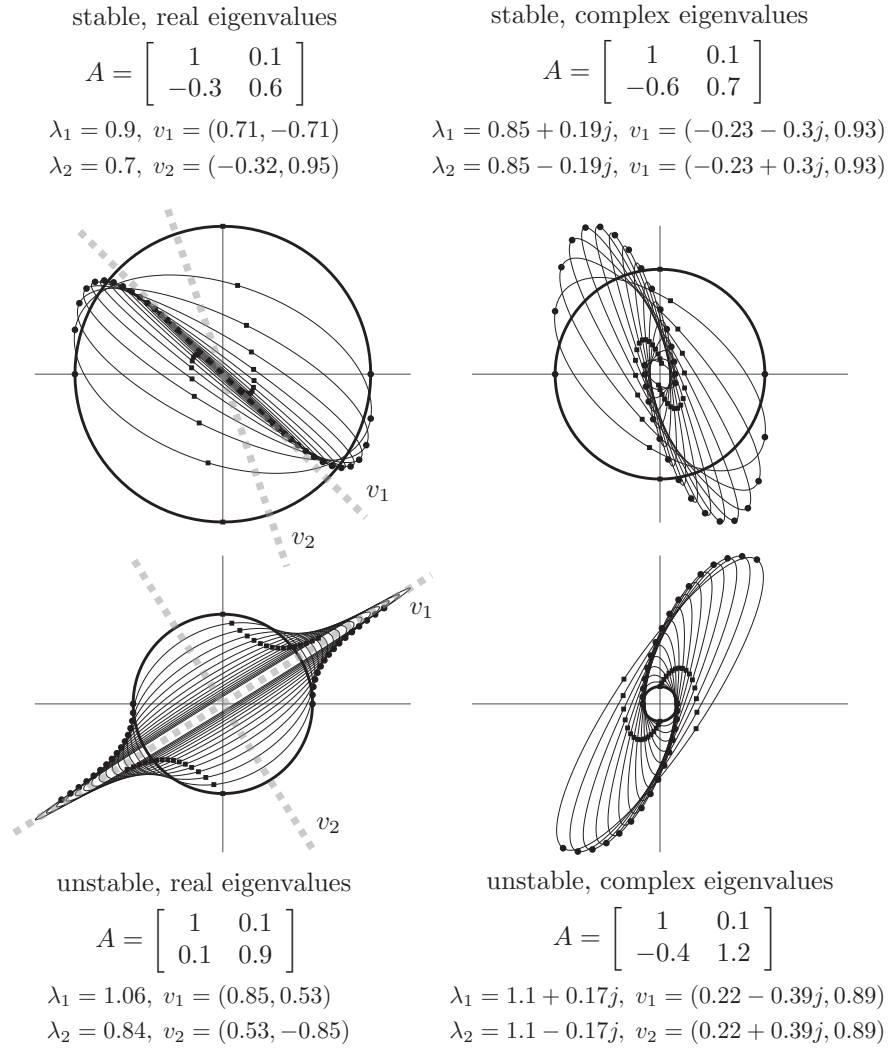
stable, real eigenvalues

$$A = \begin{bmatrix} 1 & 0.1 \\ -0.3 & 0.6 \end{bmatrix}$$

$\lambda_1 = 0.9, \ v_1 = (0.71, -0.71)$

$\lambda_2 = 0.7, \ v_2 = (-0.32, 0.95)$

stable, complex eigenvalues

$$A = \begin{bmatrix} 1 & 0.1 \\ -0.6 & 0.7 \end{bmatrix}$$

$\lambda_1 = 0.85 + 0.19j, \ v_1 = (-0.23 - 0.3j, 0.93)$

$\lambda_2 = 0.85 - 0.19j, \ v_1 = (-0.23 + 0.3j, 0.93)$

unstable, real eigenvalues

$$A = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

$\lambda_1 = 1.06, \ v_1 = (0.85, 0.53)$

$\lambda_2 = 0.84, \ v_2 = (0.53, -0.85)$

unstable, complex eigenvalues

$$A = \begin{bmatrix} 1 & 0.1 \\ -0.4 & 1.2 \end{bmatrix}$$

$\lambda_1 = 1.1 + 0.17j, \ v_1 = (0.22 - 0.39j, 0.89)$

$\lambda_2 = 1.1 - 0.17j, \ v_2 = (0.22 + 0.39j, 0.89)$

Figure E.5: A unit circle of points x in a two-dimensional space, drawn as a bold circle, iteratively mapped through A (i.e., x, Ax, A^2x, A^3x, etc.). Four points are traced through the mappings. (Top left) The eigenvalues all have less than unit magnitude, so the mapping is stable, and the points converge to the origin. The eigenvectors and eigenvalues are real. (Top right) Stable complex eigenvalues. (Bottom left) One unstable and one stable real eigenvalue. (Bottom right) Unstable complex conjugate eigenvalues.

say the system is controllable if the controls u can eventually drive the state to any $x \in \mathbb{R}^n$. Note that

$$
\begin{aligned}
x(k+2) &= Cx(k+1) + Du(k+1) \\
&= C\Big(Cx(k) + Du(k)\Big) + Du(k+1) \\
&= C^2 x(k) + CDu(k) + Du(k+1)
\end{aligned}
$$

and

$$
\begin{aligned}
x(k+3) &= Cx(k+2) + Du(k+2) \\
&= C^3 x(k) + C^2 Du(k) + CDu(k+1) + Du(k+2),
\end{aligned}
$$

etc., and finally

$$
x(k+n) = C^n x(k) + [D \ \ CD \ \ C^2 D \ \ \cdots \ \ C^{n-1}D] \begin{bmatrix} u(k+n-1) \\ \vdots \\ u(k+1) \\ u(k) \end{bmatrix}.
$$

If the $n \times nm$ matrix $[D \ \ CD \ \ C^2 D \ \ \cdots \ \ C^{n-1}D]$ satisfies

$$
\mathrm{rank}([D \ \ CD \ \ C^2 D \ \ \cdots \ \ C^{n-1}D]) = n, \tag{E.20}
$$

then the n controls at times k through $k + n - 1$ can be used to achieve any arbitrary $x(k+n)$, regardless of the initial state $x(k)$ and the drift term $C^n x(k)$. In other words, the system is controllable. The condition (E.20) is known as the **Kalman rank condition**. If the Kalman rank condition is not satisfied, then the system cannot be controlled to some states regardless of the number of timesteps.

E.7.2   Continuous-Time Control

The error dynamics for a general continuous-time control system can be written

$$
\dot{x} = Fx + Gu, \tag{E.21}
$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $F \in \mathbb{R}^{n \times n}$, and $G \in \mathbb{R}^{n \times m}$. Choosing a feedback controller $u = Kx$, where $K \in \mathbb{R}^{m \times n}$, the error dynamics become

$$
\dot{x} = (F + GK)x(k) = Bx(k).
$$

For the quadrotor, $K = [k_p \ k_d]$ and B is given in Equation (E.17).

The solution to the error dynamics differential equation $\dot{x} = Bx$ is $x(t) = e^{Bt}x(0)$, as described in Chapters 3 and 11. The error dynamics are stable (the error converges to zero) if and only if the real components of all the eigenvalues of B are negative, i.e., $\mathrm{Re}(\lambda_i) < 0$ for all i. The more negative the real components, the faster the error converges to zero.

The stability condition $\mathrm{Re}(\lambda_i) < 0$ for all i can also be derived from the fact that, for the discrete system $x(k+1) = Ax(k)$, the eigenvalues must all have magnitude less than one. To use this fact, let's approximate the differential equation $\dot{x} = Bx$ by the difference equation

$$\frac{1}{\Delta t}\Big(x(k+1) - x(k)\Big) = Bx(k)$$

for arbitrarily small $\Delta t > 0$. This equation can be rewritten as

$$x(k+1) = (I + B\Delta t)x(k).$$

For stability, the matrix $(I + B\Delta t)$ must be convergent for arbitrarily small $\Delta t > 0$, i.e., all eigenvalues must be less than unit magnitude. Since the eigenvalues of a matrix $(I + B\Delta t)$ are simply the eigenvalues of $B\Delta t$ plus one (the offset contributed by the identity matrix), the eigenvalues of $B\Delta t$ must have a negative real component for any $\Delta t > 0$, as illustrated in Figure E.6.

Figure E.7 shows examples of the quadrotor's error dynamics for different choices of k_p and k_d. Four different cases are shown: two stable cases $(\mathrm{Re}(\lambda_1), \mathrm{Re}(\lambda_2) < 0)$, one where both eigenvalues are real and one where they are complex conjugates; and two unstable cases, one where both eigenvalues are real and greater than zero and one where they are complex conjugates with a positive real component. Each case shows an initial circle of errors in the (x_1, x_2)-space and how it evolves under the dynamics $\dot{x} = Bx$.

The system (E.21) is controllable, i.e., it can be driven to any state $x \in \mathbb{R}^n$ in finite time, if it satisfies the Kalman rank condition

$$\mathrm{rank}(\begin{bmatrix} G & FG & F^2G & \cdots & F^{n-1}G \end{bmatrix}) = n.$$

E.8   Glossary

broad matrix: See *fat matrix*.

characteristic polynomial: The characteristic polynomial $p(s)$ of a square matrix A is $\det(sI - A)$, and the roots of the characteristic polynomial are the eigenvalues of A.
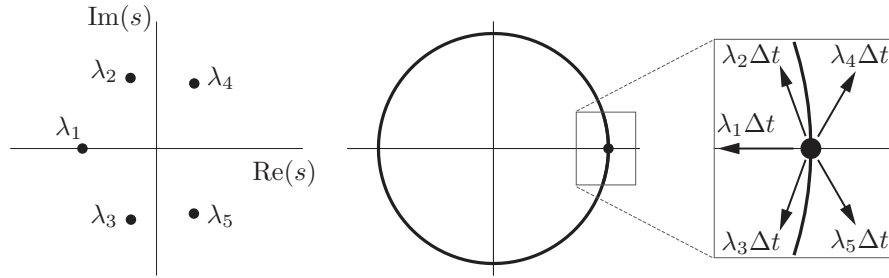
Figure E.6: Left: Example eigenvalues of a 5×5 matrix B illustrated in the complex plane. The real eigenvalue λ_1 and the complex conjugates $\lambda_{2,3}$ have negative real components, while the complex conjugates $\lambda_{4,5}$ have positive real components. Middle: The five eigenvalues of the 5×5 identity matrix I are all at 1. Also shown is the unit circle in the complex plane. Right: Zooming in near $s = 1$, we see the eigenvalues of $I + B\Delta t$. The eigenvalues of $B\Delta t$ are $\lambda_1 \Delta t$, $\lambda_2 \Delta t$, etc. As Δt shrinks toward zero, the eigenvalues $1 + \lambda_i \Delta t$ are inside the unit circle if and only if $\text{Re}(\lambda_i) < 0$. In this example, the eigenvalues $1 + \lambda_i \Delta t$ for $i = 1, 2, 3$ are inside the unit circle for infinitesimal $\Delta t > 0$ but the eigenvalues for $i = 4, 5$ are outside the unit circle.

codomain: Viewing an $m \times n$ matrix as mapping $\mathbb{R}^n$ to $\mathbb{R}^m$, the codomain is $\mathbb{R}^m$.

column space: See *range*.

condition number: The ratio of the largest and smallest singular values of a full-rank matrix A, $\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$.

convergent matrix: A square matrix A is convergent if A^k converges to the zero matrix as k goes to infinity.

determinant: The determinant of a square matrix A is a polynomial of the elements of A satisfying $\det(A) = \lambda_1 \lambda_2 \ldots \lambda_n$, where $\{\lambda_i\}$ are eigenvalues of A.

diagonal matrix: All non-diagonal elements a_{ij}, $i \neq j$, are equal to zero for a diagonal matrix A.

diagonalizable: A square matrix $A \in \mathbb{R}^{n \times n}$ is diagonalizable if there exists an orthogonal matrix $Q \in \mathbb{R}^{n \times n}$ such that $A = QDQ^{\text{T}}$, where $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix.

domain: Viewing an $m \times n$ matrix as mapping $\mathbb{R}^n$ to $\mathbb{R}^m$, the domain is $\mathbb{R}^n$.

eigenvalue: A complex number $\lambda \in \mathbb{C}$ is an eigenvalue of $A \in \mathbb{R}^{n \times n}$ if $Av = \lambda v$, where v is an eigenvector.

stable, real eigenvalues

$$B = \begin{bmatrix} 0 & 1 \\ -0.2 & -1.1 \end{bmatrix}$$

$\lambda_1 = -0.87, \ v_1 = (0.75, -0.66)$

$\lambda_2 = -0.23, \ v_2 = (0.97, -0.22)$

stable, complex eigenvalues

$$B = \begin{bmatrix} 0 & 1 \\ -1.5 & -1.1 \end{bmatrix}$$

$\lambda_1 = -0.55 + 1.09j, \ v_1 = (-0.28 - 0.57j, 0.77)$

$\lambda_2 = -0.55 - 1.09j, \ v_1 = (-0.28 + 0.57j, 0.77)$

unstable, real eigenvalues

$$B = \begin{bmatrix} 0 & 1 \\ -0.1 & 0.7 \end{bmatrix}$$

$\lambda_1 = 0.5, \ v_1 = (0.89, 0.45)$

$\lambda_2 = 0.2, \ v_2 = (0.98, 0.2)$

unstable, complex eigenvalues

$$B = \begin{bmatrix} 0 & 1 \\ -2 & 1 \end{bmatrix}$$

$\lambda_1 = 0.5 + 1.32j, \ v_1 = (0.20 - 0.54j, 0.82)$

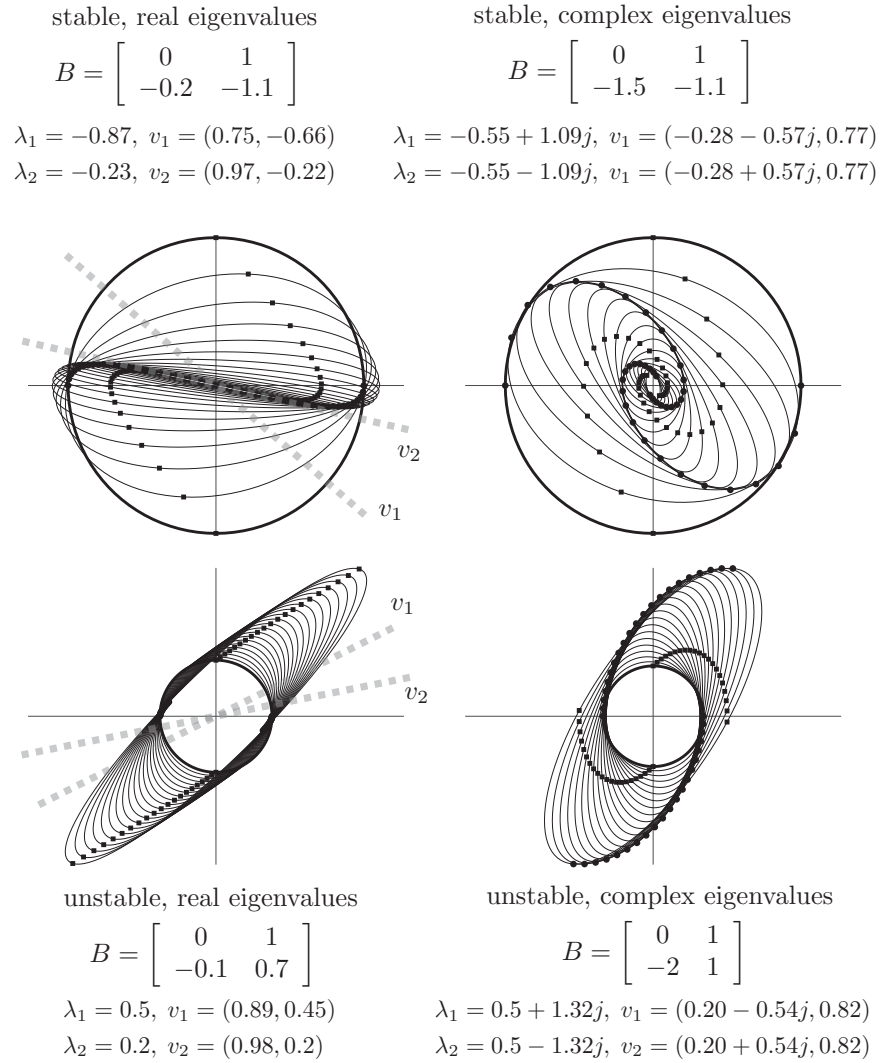$\lambda_2 = 0.5 - 1.32j, \ v_2 = (0.20 + 0.54j, 0.82)$

Figure E.7: A unit circle of points x in a two-dimensional space, drawn as a bold circle, evolving according to $\dot{x} = Bx$. The evolving circle is plotted at fixed time intervals, and four points are traced as they evolve. (Top left) The eigenvalues both have negative real components, so the differential equation is stable. The eigenvectors and eigenvalues are real. (Top right) Stable complex eigenvalues. (Bottom left) Two unstable real eigenvalues. (Bottom right) Unstable complex conjugate eigenvalues.

eigenvector: An eigenvector $v \in \mathbb{C}^n$ of a square matrix $A \in \mathbb{R}^{n \times n}$ satisfies $Av = \lambda v$, where λ is an eigenvalue.

fat matrix: A matrix $A \in \mathbb{R}^{m \times n}$ is fat if it has more columns than rows $(n > m)$.

full rank: A matrix $A \in \mathbb{R}^{m \times n}$ is full rank if its rank equals $\min(m, n)$.

identity matrix: A diagonal matrix with all elements on the diagonal equal to one.

ill-conditioned matrix: A matrix with a large condition number.

image: See *range*.

inverse (of a square matrix): The inverse A^{-1} of a square matrix A satisfies $AA^{-1} = A^{-1}A = I$. The inverse exists only if A is full rank.

kernel: See *null space*.

left inverse: The left inverse A_{L}^{-1} of a matrix A satisfies $A_{\mathrm{L}}^{-1}A = I$.

left-singular vectors: The columns of the U matrix in the singular-value decomposition $A = U\Sigma V^{\mathrm{T}}$.

linear span: The linear span of a set of vectors a_1, a_2, ..., a_n is all vectors of the form $\Sigma_i k_i a_i$, where each k_i is a real number.

linearly independent: A vector v is linearly independent of a set of vectors $a_1, \ldots, a_n$ if there are no real coefficients k_i satisfying $v = \Sigma_i k_i a_i$.

non-square matrix: A matrix $A \in \mathbb{R}^{m \times n}$ is non-square if $m \neq n$.

null space: The null space of a matrix A is the space of all vectors x such that $Ax = 0$.

nullity: The nullity of a matrix A is the dimension of its null space.

orthogonal group $O(n)$: The set of all $n \times n$ orthogonal matrices.

orthogonal matrix: A square matrix whose columns are unit vectors that are orthogonal to each other. (The rows are also unit vectors orthogonal to each other.) The determinant of an orthogonal matrix is $+1$ or -1.

orthonormal vectors: Two vectors of the same dimension are orthonormal if they are unit vectors (normal) and orthogonal (i.e., their dot product is zero).

poorly-conditioned matrix: See *ill-conditioned matrix*.

positive-definite matrix: A symmetric matrix whose eigenvalues are all positive.

pseudoinverse: The pseudoinverse $A^{\dagger}$ of a matrix A satisfies $AA^{\dagger}A = A$, $A^{\dagger}AA^{\dagger} = A^{\dagger}$, $AA^{\dagger}$ symmetric, and $A^{\dagger}A$ symmetric.

range: For a matrix $A \in \mathbb{R}^{m \times n}$, the range is the space of all Ax for all $x \in \mathbb{R}^n$.

rank: The rank of a matrix A is equal to the dimension of its range.

rank deficient: See *singular*.

right inverse: The right inverse A_R^{-1} of a matrix A satisfies $AA_\mathrm{R}^{-1} = I$.

right-singular vectors: The columns of the V matrix in the singular-value decomposition $A = U\Sigma V^\mathrm{T}$.

rotation matrix: An element of the special orthogonal group $SO(3)$, representing a rigid-body orientation or rotation.

$se(3)$: The set of all 4×4 real matrices with the top left 3×3 submatrix an element of $so(3)$ and the bottom row all zeros. The matrix representation $[\mathcal{V}]$ of a twist $\mathcal{V} \in \mathbb{R}^6$ is an element of $se(3)$.

singular: A matrix $A \in \mathbb{R}^{m \times n}$ is singular if its rank is less than the smaller of m and n.

singular value: A singular value σ_i of a matrix A is the nonnegative amplification of a unit right-singular vector v_i of the matrix A when mapping it through the matrix A, i.e., $\|Av_i\| = \sigma_i$. See *singular-value decomposition*.

singular-value decomposition (SVD): The singular-value decomposition of a matrix $A \in \mathbb{R}^{m \times n}$ is $A = U\Sigma V^\mathrm{T}$, where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ is a matrix of zeros other than the singular values σ_i along the diagonal.

singularity: If a matrix A depends on a variable or variables θ, and the rank of $A(\theta)$ is full at some values of θ but less than full at other values of θ, the values of θ where $A(\theta)$ is less than full rank are called singularities.

skew-symmetric matrix: A matrix A is skew symmetric if $A = -A^\mathrm{T}$.

$so(3)$: The set of all 3×3 real skew-symmetric matrices. The matrix representation $[\omega]$ of an angular velocity $\omega \in \mathbb{R}^3$ is an element of $so(3)$.

special Euclidean group $SE(3)$: The set of all 4×4 real matrices with the top left 3×3 submatrix an element of $SO(3)$ and the bottom row equal to $[0\ 0\ 0\ 1]$. Elements of $SE(3)$ are used to represent rigid-body configurations and displacements.

special orthogonal group $SO(3)$: The set of all 3×3 real orthogonal matrices with determinant $+1$. Elements of $SO(3)$ are used to represent rigid-body orientations and rotations.

square matrix: A matrix $A \in \mathbb{R}^{m \times n}$ is square if it has an equal number of rows and columns ($m = n$).

symmetric matrix: A matrix A is symmetric if $A^\mathrm{T} = A$.

tall matrix: A matrix $A \in \mathbb{R}^{m \times n}$ is tall if it has more rows than columns $(m > n)$.

trace: The sum of the diagonal elements of a square matrix.

transformation matrix: An element of the special Euclidean group $SE(3)$, representing a rigid-body configuration or displacement.

transpose: The transpose A^{T} of a matrix A has the columns of A as its rows.

wide matrix: See *fat matrix*.

Bibliography

[1] C. D. Meyer. *Matrix analysis and applied linear algebra*. SIAM, 2000.

[2] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. Version November 15, 2012. Available online.

[3] G. Strang. *Introduction to Linear Algebra, Fourth Edition*. Wellesley-Cambridge Press, 2009.