Linear Algebra and Robot Modeling

Nathan Ratliff

Apr 8, 2014

Abstract

Linear algebra is fundamental to robot modeling, control, and optimization. This document reviews some of the basic kinematic equations and uses them to motivate an SVD-centric geometric perspective on linear algebra. This perspective illuminates the underlying structure and behavior of linear maps and simplifies analysis, especially for reduced rank matrices. We later review some properties of multidimensional quadratic forms that are central to optimal control. Most of this material should already be familiar to the reader; this document explores some specifics and offers a potentially unique intuition oriented perspective.

1 Basic kinematic equations

Let $\mathcal{C} \subset \mathbb{R}^d$ be a configuration space. For instance, the configuration space may be a space of valid joint angles for a robot manipulator. Consider as a running example the differentiable map mapping a point in the configuration space (a particular set of joint angles) to the three-dimensional location of the manipulator's fingertip. We denote this *forward kinematics* map abstractly as $\phi : \mathcal{C} \to \mathbb{R}^3$.

This map is no different from any other multidimensional differentiable mapping from calculus, so we can ask the typical calculus questions such as how does the output change with changes in the input, etc. Kinematic equations are none other than equations relating derivatives of the inputs to derivatives of the outputs.

Two common kinematic equations are

$$\dot{\boldsymbol{x}} = \boldsymbol{J}_\phi \dot{\boldsymbol{q}} \quad \text{and} \quad \ddot{\boldsymbol{x}} = \boldsymbol{J}_\phi \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_\phi \dot{\boldsymbol{q}}, \tag{1}$$

where $\boldsymbol{J}_\phi = \frac{\partial \phi}{\partial \boldsymbol{q}}$ is the Jacobian (total derivative) of ϕ. These equations are easy to write down, especially when referencing tables of derivative formulas in an Advanced Calculus text, but in robotics we need a strong intuitive understanding of what they mean. For instance, two questions that may pop to mind are

1. How does that first equation relate to the equation $\delta \boldsymbol{x} = \boldsymbol{J}_\phi \delta \boldsymbol{q}$?
2. What does $\dot{\boldsymbol{J}}_\phi$ actually mean, and how do we compute it?

1.1 Interpreting time derivatives

Whenever an equation uses time derivatives, such as $\dot{\boldsymbol{q}}$, there is an implicit assumption that $\boldsymbol{q}$ is really a trajectory $\boldsymbol{q} : [0, T] \to \mathbb{R}^d$. The time derivatives give the position $\boldsymbol{q} = \boldsymbol{q}(t)$, velocity $\dot{\boldsymbol{q}} = \frac{d}{dt}\boldsymbol{q}(t)$, and acceleration $\ddot{\boldsymbol{q}} = \frac{d}{dt}\dot{\boldsymbol{q}}(t)$ of the trajectory at some time t.

Thus, the equation $\dot{\boldsymbol{x}} = \boldsymbol{J}_\phi \dot{\boldsymbol{q}}$ relates how the velocity $\dot{\boldsymbol{q}}$ in the system's configuration space relates to the velocity of the point on the end-effector in Cartesian space. Concretely, using our manipulator example, it tells us how joint velocities (the rate at which each joint is changing over time) relate to the finger tip's velocity (the rate at which the Cartesian dimensions of the fingertip are changing over time). The implicit assumption here is always that the system is moving along some trajectory $\boldsymbol{q}(t)$, and that that trajectory gives us the time variable that allows us to address questions of how the system evolves over time.

Now given that q refers implicitly to an underlying trajectory, we can now interpret what $\dot{\boldsymbol{J}}_\phi$ means. As the system moves along its trajectory from $\boldsymbol{q}(t)$ to a point dt in the future $\boldsymbol{q}(t + dt)$, the associated Jacobian changes slightly as well since it's a non-constant function of t. $\dot{\boldsymbol{J}}_\phi$ gives explicitly the rate at which the Jacobian is changing with time:

$$\dot{\boldsymbol{J}}_\phi = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \Big(\boldsymbol{J}_\phi(q(t + \Delta t)) - \boldsymbol{J}_\phi(q(t)) \Big) \tag{2}$$

We usually calculate this quantity numerically by taking a finite-difference between the Jacobian now and the Jacobian we find a little bit in the direction of the current velocity. That approximation can be derived from the above using a first-order Taylor expansion and a finite time step:

$$\dot{\boldsymbol{J}}_\phi \approx \frac{1}{\Delta t} \Big(\boldsymbol{J}_\phi(\boldsymbol{q}(t + \Delta t)) - \boldsymbol{J}_\phi(\boldsymbol{q}(t)) \Big) \tag{3}$$

$$= \frac{1}{\Delta t} \Big(\boldsymbol{J}_\phi(\boldsymbol{q}(t) + \Delta t \dot{\boldsymbol{q}})) - \boldsymbol{J}_\phi(\boldsymbol{q}(t)) \Big) . \tag{4}$$

1.2 Tangent vectors and co-tangent vectors

How does the expression $\delta\boldsymbol{x} = \boldsymbol{J}_\phi \delta\boldsymbol{q}$ differ from the expression above relating velocities $\dot{\boldsymbol{x}} = \boldsymbol{J}_\phi \dot{\boldsymbol{q}}$? Notationally, $\delta\boldsymbol{q}$ denotes a small displacement of $\boldsymbol{q}$ which is qualitatively similar to a velocity vector, but depending on whether you're an engineer with a tool belt full of math tricks or a mathematical purist the two expressions can range from being "slightly different but heuristically the same" to being "fundamentally different". Clearly, it's superficially similar because derivatives work in very structured ways, but this equation doesn't assume that there's any underlying trajectory. The quantity $\delta\boldsymbol{q}$ should be thought of as a *small perturbation* in $\boldsymbol{q}$, and the equation tells us how small perturbations in $\boldsymbol{q}$ result in small perturbations of $\boldsymbol{x}$.

There are technical details that we won't worry about in this class, but rigorous treatments of these ideas can be found in textbooks on differential geometry. There, $\dot{\boldsymbol{q}}$ is known as a *tangent* vector, and these small perturbations

$\delta \boldsymbol{q}$ are known as *co-tangent* vectors. The former generalizes tangents to curves across a manifold, and the latter generalizes gradients of scalar functions defined on the manifold.

Within robotics, rigor is often secondary to physical intuition, so definitions and manipulations of these ideas are less stringent. The bottom line is that $\dot{\boldsymbol{q}}$ always refers to a time derivative of an underlying trajectory, whereas $\delta \boldsymbol{q}$ will simply denote a perturbation (small movement of) the point $\boldsymbol{q}$ in space with no notion of *time*.

We'll often use $\delta \boldsymbol{q}$ to also represent a finite-sized perturbation away from some approximation point $\boldsymbol{q}$, which lets us write the first-order Taylor expansion of a scalar function $c : \mathcal{C} \to \mathbb{R}$ as

$$c(\boldsymbol{q} + \delta \boldsymbol{q}) \approx c(\boldsymbol{q}) + \nabla c(\boldsymbol{q})^T \delta \boldsymbol{q}. \tag{5}$$

In this notation, the expression $\delta \boldsymbol{x} = (\boldsymbol{x} + \delta \boldsymbol{x}) - \boldsymbol{x} \approx \boldsymbol{J}_\phi \delta \boldsymbol{q}$ can be viewed as a first-order Taylor expansion of the map ϕ. Following this line of thought, we can also expand the approximation to a second order and write[1]

$$\delta \boldsymbol{x} \approx \boldsymbol{J}_\phi \delta \boldsymbol{q} + \frac{1}{2} \left[\delta \boldsymbol{q}^T \nabla^2 \phi_k(\boldsymbol{q}) \delta \boldsymbol{q} \right]_k \tag{6}$$

This notion of second-order expansion doesn't make sense for $\dot{\boldsymbol{x}} = \boldsymbol{J}_\phi \dot{\boldsymbol{q}}$ since we always treat this latter equation as an exact differential expression relating time-rates of change.

2 A geometric perspective on linear algebra

Linear algebra is fundamental to control and optimization. This section reviews some of the most important properties of linear maps from the perspective of the Singular Value Decomposition (SVD). If we start from the notion that every matrix has an SVD, the structure and behavior of the matrix becomes immediately clear and algebraic manipulations of even reduced rank matrices are straightforward. We assume familiarity with the basic linear algebraic concepts and focus here on understanding the underlying geometry of linear maps.

2.1 Linear maps are bijections between fundamental spaces

The *row space* and *column space* are fundamental to a linear map. We'll see in this section that a linear map forms a bijection[2] between between these two spaces, and that all components in the orthogonal compliment to these spaces are simply removed when necessary. These orthogonal compliments are known as the left and right *null spaces*, respectively.

[1] For those familiar with tensor notation and the Einstein summation convention, this second term can be more compactly written as $\frac{1}{2} \left(\partial_{ij} \phi_k \right) \delta \boldsymbol{q}^i \delta \boldsymbol{q}^j$.

[2] A bijection is formally a *one-to-one* and *onto* function. One may view it as a full pairing between points in the domain and range. Every domain point $\boldsymbol{x}$ has an corresponding range point $\boldsymbol{y}$ and vice versa under the bijection.

Usually, the row and column spaces are defined as the space spanning the rows and the space spanning the columns. Those definitions, though true, aren't very insightful. This section shows that the SVD provides a nice geometric view of what these spaces are and how the linear map operates on them. This decomposition gives geometric insight into the fundamental nature of linear algebra.

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a matrix representing a linear map from $\mathbb{R}^n$ to $\mathbb{R}^m$. We know from what is sometimes referred to as the *fundamental theorem of linear algebra* that $\boldsymbol{A}$ has a Singular Value Decomposition (SVD) of the form $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is a diagonal matrix of singular values $\sigma_1, \ldots, \sigma_k$. We don't assume that $\boldsymbol{A}$ is full rank, so k may be less than both m and n. Since $\boldsymbol{S}$ is a non-square diagonal matrix with only $k \leq m, n$ nonzero entries, we can better reveal its structure by writing it as

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} \\ \mathbf{0} \end{pmatrix} \boldsymbol{\Sigma} \begin{pmatrix} \boldsymbol{I} & \mathbf{0} \end{pmatrix} \tag{7}$$

where the $\mathbf{0}$ matrices are appropriately sized and $\boldsymbol{\Sigma} = \mathbf{diag}(\sigma_1, \ldots, \sigma_k)$.

Since $\boldsymbol{S}$ as shown in Equation 7 decomposes as a square diagonal matrix $\boldsymbol{\Sigma}$ with zero buffers on either side, we can further decompose both $\boldsymbol{U}$ and $\boldsymbol{V}$ into

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U}_{/\!/} & \boldsymbol{U}_{\perp} \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{V} = \begin{pmatrix} \boldsymbol{V}_{/\!/} & \boldsymbol{V}_{\perp} \end{pmatrix}, \tag{8}$$

where the columns with the subscript $\perp$ are the ones annihilated by the zeros and the columns with the subscript $/\!/$ are those that remain. This decomposition allows us to rewrite the SVD as

$$\begin{aligned} \boldsymbol{A} &= \begin{bmatrix} \boldsymbol{U}_{/\!/} & \boldsymbol{U}_{\perp} \end{bmatrix} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{bmatrix} \boldsymbol{V}_{/\!/}^T \\ \boldsymbol{V}_{\perp}^T \end{bmatrix} && (9) \\ &= \left(\begin{bmatrix} \boldsymbol{U}_{/\!/} & \boldsymbol{U}_{\perp} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} \right) \boldsymbol{\Sigma} \left(\begin{bmatrix} \boldsymbol{V}_{/\!/} & \boldsymbol{V}_{\perp} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} \right)^T && (10) \\ &= \boldsymbol{U}_{/\!/} \boldsymbol{\Sigma} \boldsymbol{V}_{/\!/}^T. && (11) \end{aligned}$$

It's fairly easy to show that, in terms of this decomposition, $\mathbf{span}(\boldsymbol{V}_{/\!/})$ is the column space, $\mathbf{span}(\boldsymbol{V}_{\perp})$ is the right null space, $\mathbf{span}(\boldsymbol{U}_{/\!/})$ is the row space, and $\mathbf{span}(\boldsymbol{U}_{\perp})$ is the left null space (which we'll see below is the null space of the natural generalized inverse).

The last expression in Equation 11, known as the *thin* SVD, reveals the fundamental structure of any (possibly reduced rank) matrix. Some, depending on background, find it more insightful to view that expression as

$$\boldsymbol{A} = \boldsymbol{U}_{/\!/} \boldsymbol{\Sigma} \boldsymbol{V}_{/\!/}^T = \sum_{i=1}^{k} \sigma_i \ \boldsymbol{u}_i \boldsymbol{v}_i^T, \tag{12}$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the k columns of $\boldsymbol{U}_{/\!/}$ and $\boldsymbol{V}_{/\!/}$, respectively.

Geometrically, this expression says that all rank k matrices are simultaneously a correspondence between a k-dimensional orthogonal basis in the domain space $\mathcal{B}_\mathcal{V} = \{\boldsymbol{v}_i\}_{i=1}^k$ and a k-dimensional orthogonal basis in the range space $\mathcal{B}_\mathcal{U} = \{\boldsymbol{u}_i\}_{i=1}^k$ with a rule for how vectors should be stretched or compressed (or reflected when $\sigma_i < 0$) along those dimensions. (We refer to these bases below as simply the matrices $\boldsymbol{V}_{/\!/}$ and $\boldsymbol{U}_{/\!/}$, respectively.)

It's useful to think about each term in the expansion in relation to its corresponding *endomorphism*[3] $\sum_i \sigma_i \, \boldsymbol{u}_i \boldsymbol{u}_i^T$. Each term of this endomorphism acts on a vector $\boldsymbol{x}$ to find its component $(\boldsymbol{u}_i^T \boldsymbol{x})\boldsymbol{u}_i$ along the basis element $\boldsymbol{u}_i$ and stretch it by σ_i. Since each basis element is orthogonal to all others, this endomorphism simply stretches, compresses, and/or reflects the vector independently along the given basis directions.

Equation 12, therefore, shows that all matrices have this fundamental behavior. Except in this case, when the domain and range spaces differ, we need to both decompose the vector $\boldsymbol{x}$ in terms of the domain basis and *simultaneously map* that decomposition onto the corresponding basis of the range space. Once that connection between the (sub)spaces is established the matrix can apply its underlying operation defined by its singular values σ_i.

The following equivalent expressions illustrate the underlying behavior of the linear map from multiple perspectives:

$$\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T = \underbrace{\left(\boldsymbol{U}_{/\!/}\boldsymbol{V}_{/\!/}^T\right)\left(\boldsymbol{V}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T\right)}_{\text{Stretch then map}} = \underbrace{\left(\boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{U}_{/\!/}^T\right)\left(\boldsymbol{U}_{/\!/}\boldsymbol{V}_{/\!/}^T\right)}_{\text{Map then stretch}}, \tag{13}$$

Each gives a subtly different way of thinking about how the matrix $\boldsymbol{A}$ operates. The first says that we can think of $\boldsymbol{A}$ physically as a stretching/squishing of the domain space space followed by an incompressible mapping between orthogonal bases (simply mapping each domain basis element to its corresponding range basis element), whereas the second says we can equivalently view $\boldsymbol{A}$ as first a mapping between basis elements followed by a stretching of the space. It's natural then to think of $\boldsymbol{A}$'s action holistically as simply an association between the two k-dimensional subspaces defined by $\boldsymbol{U}_{/\!/}\boldsymbol{V}_{/\!/}^T$ and a corresponding stretching/squishing/reflection of that unified space.

2.2 A natural generalized inverse

Note that the basis-correspondence mappings of the form $\boldsymbol{U}_{/\!/}\boldsymbol{V}_{/\!/}^T = \sum_{i=1}^k \boldsymbol{u}_i \boldsymbol{v}_i^T$ are all between k-dimensional subspaces, not the full space. What happens to the dimensions of $\boldsymbol{x}$ orthogonal that space in the domain in the expression $\boldsymbol{A}\boldsymbol{x} = \left(\boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T\right)\boldsymbol{x}$? They vanish![4] There aren't enough singular values to represent those dimensions, so simply by the basic structure of a linear map, dimensions orthogonal to the fundamental k dimensional subspaces are removed.

[3] An endomorphism is a mapping from a space back onto the same space.

[4] More explicitly, the operator $\boldsymbol{V}_{/\!/}\boldsymbol{V}_{/\!/}^T$ projects a vector onto the space $\mathbf{span}(\boldsymbol{V}_{/\!/})$, and $\boldsymbol{V}_\perp \boldsymbol{V}_\perp^T$ projects onto $\mathbf{span}(\boldsymbol{V}_\perp)$, so we can always decompose a domain vector $\boldsymbol{x}$ as $\boldsymbol{x} = \left(\boldsymbol{V}_{/\!/}\boldsymbol{V}_{/\!/}^T\right)\boldsymbol{x} +$

In other words, the mapping between the subspaces is bijective, and any components orthogonal to those spaces cannot be represented by the linear map. Thus, if $\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T$ is the forward map implementing a forward mapping between fundamental spaces and removing any domain element orthogonal to the column space then it reasons to say that the opposite mapping procedure which implements the inverse of that bijection between fundamental space and removes any component orthogonal to the row space is a natural generalized inverse for this map. This generalized inverse is given explicitly by the map

$$\boldsymbol{A}^\dagger = \boldsymbol{V}_{/\!/}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T. \tag{14}$$

It's straightforward to show that $\boldsymbol{A}^\dagger$ gives the exact inverse on the bijection between the fundamental k-dimensional subspaces, but any component orthogonal to $\boldsymbol{V}_{/\!/}$ is removed. This expression is exactly the Moore-Penrose pseudoinverse:

$$\boldsymbol{A}^\dagger = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}, \tag{15}$$

as can be shown simply by expanding each $\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T$. In other words, by construction, we've shown that the Moore-Penrose pseudoinverse is a natural generalization of the notion of inverse. Prescriptively, it says that between the domain and range spaces the right thing to do is perform either the forward or inverse bijection between the fundamental k-dimensional subspaces, and simply remove any components orthogonal to those spaces. For the forward map, we remove any component orthogonal to the column space and perform the forward bijection. And for the inverse map, we remove any component orthogonal to the row space and perform the backward bijection.

2.3 Using the SVD to solve reduced rank linear equations

Let $\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T \in \mathbb{R}^{m\times n}$ be any rank k matrix as above. $\boldsymbol{A}$ may be reduced rank, making k smaller than either m or n. Solving the equation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

fully for the entire space of valid solutions is straightforward using this decomposition as we show here:

$$\begin{aligned} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \Rightarrow \quad & \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T\boldsymbol{x} = \boldsymbol{b} \\ \Rightarrow \quad & \boldsymbol{V}_{/\!/}^T\boldsymbol{x} = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T\boldsymbol{b}. \end{aligned} \tag{16}$$

$\left(\boldsymbol{V}_\perp\boldsymbol{V}_\perp^T\right)\boldsymbol{x}$. Using that decomposition,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T\left[\left(\boldsymbol{V}_{/\!/}\boldsymbol{V}_{/\!/}^T\right)\boldsymbol{x} + \left(\boldsymbol{V}_\perp\boldsymbol{V}_\perp^T\right)\boldsymbol{x}\right] \\ &= \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T\boldsymbol{x}, \end{aligned}$$

since $\boldsymbol{V}_{/\!/}^T\boldsymbol{V}_\perp = \boldsymbol{0}$ by construction.

$\boldsymbol{x}$ can always be decomposed as a linear combination of the columns $\boldsymbol{V}_{/\!/}$ and a linear combination of the columns $\boldsymbol{V}_{\perp}$ as in $\boldsymbol{x} = \boldsymbol{V}_{/\!/}\alpha + \boldsymbol{V}_{\perp}\beta$, where α and β are the associated coefficients. Doing so shows that the left hand side of the expression in Equation 16 reduces to

$$\begin{aligned}\boldsymbol{V}_{/\!/}^T\boldsymbol{x} &= \boldsymbol{V}_{/\!/}^T\left(\boldsymbol{V}_{/\!/}\alpha + \boldsymbol{V}_{\perp}\beta\right)\\ &= \left(\boldsymbol{V}_{/\!/}^T\boldsymbol{V}_{/\!/}\right)\alpha + \left(\boldsymbol{V}_{/\!/}^T\boldsymbol{V}_{\perp}\right)\beta\\ &= \alpha.\end{aligned}$$

In other words, the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ constrains $\boldsymbol{x}$ only through α, which is given by

$$\alpha = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T\boldsymbol{b}. \tag{17}$$

The full solution is, therefore,

$$\boldsymbol{x} = \boldsymbol{V}_{/\!/}\alpha + \boldsymbol{V}_{\perp}\beta = \underbrace{\left(\boldsymbol{V}_{/\!/}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T\right)}_{\boldsymbol{A}^{\dagger}}\boldsymbol{b} + \boldsymbol{V}_{\perp}\beta \tag{18}$$

for any $(n-k)$-dimensional coefficient vector β. As indicated, the first term is just the well known pseudoinverse solution, and the second term is an element of the null space spanned by $\boldsymbol{V}_{\perp}$.

This notation is somewhat different than commonly used to describe these solutions (raw matrix operations such as in Equation 15), but even computationally, expressions derived through applications of SVDs can be easily and robustly implemented, too, since matrix libraries make SVDs computations readily available in practice. The trade-off is robustness, stability, and geometric insight (SVD) for computational speed (raw matrix operations).

2.4 Solving the system without algebra

Consider the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ can be any possibly reduced rank matrix. Then, as discussed above, the decomposition $\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T$ tells us that $\boldsymbol{A}$ transports specifically only the portion of $\boldsymbol{x}$ lying within $\mathbf{span}(\boldsymbol{V}_{/\!/})$ to a point in the space $\mathbf{span}(\boldsymbol{U}_{/\!/})$. That means that all vectors of the form

$$\boldsymbol{z}(\beta) = \boldsymbol{x}_0 + \boldsymbol{V}_{\perp}\beta \tag{19}$$

are transported to the same point under $\boldsymbol{A}$. In particular, when $\boldsymbol{x}_0$ lies on the column space $\mathbf{span}(\boldsymbol{V}_{/\!/})$, i.e. $\boldsymbol{x}_0 = \boldsymbol{V}_{/\!/}\alpha$, then we know where $\boldsymbol{x}_0$ ends up under the fundamental internal bijection of $\boldsymbol{A}$ and that means we know where all points $\boldsymbol{z}(\beta)$ end up.

Thus, to solve the system, we just need to know which point in $\mathbf{span}(\boldsymbol{V}_{/\!/})$ maps to $\boldsymbol{b} \in \mathbf{span}(\boldsymbol{U}_{/\!/})$. Since the forward map takes the coordinates of the point $\boldsymbol{x}$ in the domain basis $\boldsymbol{V}_{/\!/}$, inflates them by the factors σ_i, and applies them directly to the range basis $\boldsymbol{U}_{/\!/}$, the inverse operation must just do the opposite. It

should take the coefficients in range basis $\boldsymbol{U}_{/\!/}$, shrink them by inverse factors $\frac{1}{\sigma_i}$, and apply them to domain basis $\boldsymbol{V}_{/\!/}$. Thus (as we derived above), if the forward map is $\boldsymbol{A} = \boldsymbol{U}_{/\!/}\boldsymbol{\Sigma}\boldsymbol{V}_{/\!/}^T$, the inverse map between fundamental spaces must be $\boldsymbol{A}^\dagger = \boldsymbol{V}_{/\!/}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T$ (specifically, $\boldsymbol{U}_{/\!/}^T$ finds the components in the k-dimensional orthogonal basis for $\mathbf{span}(\boldsymbol{U}_{/\!/})$, $\boldsymbol{\Sigma}^{-1}$ scales those components, and $\boldsymbol{V}_{/\!/}$ applies the resulting components directly to the k orthogonal basis elements of $\mathbf{span}(\boldsymbol{V}_{/\!/})$).

Given this intuition, we're now equipped to simply write down the full linear space solution to the system directly:

$$\boldsymbol{x}^* = \left(\boldsymbol{V}_{/\!/}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}_{/\!/}^T\right)\boldsymbol{b} + \boldsymbol{V}_\perp\beta \tag{20}$$

for any $\beta \in \mathbb{R}^k$.

Note that this argument can also be used to solve, in the least squares sense, the system when $\boldsymbol{b}$ doesn't actually lie within $\mathbf{span}(\boldsymbol{U}_{/\!/})$. In this case, we just need to find which element of $\mathbf{span}(\boldsymbol{V}_{/\!/})$ gets mapped onto the orthogonal projection of $\boldsymbol{b}$ onto $\mathbf{span}(\boldsymbol{U}_{/\!/})$, which is just the point defined by the coordinates of $\boldsymbol{b}$ in basis $\boldsymbol{U}_{/\!/}$. Considering only these coordinates, the above argument still unfolds in the same way and the solution doesn't change. In other words, we can use a geometric argument to show that that Equation 20 also solves the least squares problem.

3 Quadratic forms and their manipulation

Quadratic forms are important tools in optimization and control because of their close connection to linear systems. Above, for instance, we saw that the basic structure of a linear map encodes an implicit least squares problem, which itself may be viewed as a quadratic objective function.

This section reviews very briefly some of the basic rules governing quadratic forms that should ideally be understood at an intuitive level. These rules relate how quadratics combine with one another and how they behave under linear transformation. We'll state the rules here without proof. They can all be derived by calculation.

1. **Adding quadratics always gives another quadratic.** $Q_1(\boldsymbol{x}) + Q_2(\boldsymbol{x})$ is a quadratic over $\boldsymbol{x}$, and $Q_x(\boldsymbol{x}) + Q_q(\boldsymbol{q})$ is a quadratic function defined jointly over $\boldsymbol{x}$ and $\boldsymbol{q}$. Generally, adding any two quadratic functions of any collection of variables gives a joint quadratic over the union of those variables. Calculating the coefficients can be tedious, but we always know that it'll still be a quadratic.

2. **Linearly transforming a quadratic gives another quadratic.** If $Q(\boldsymbol{x})$ is a quadratic and $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{q} + \boldsymbol{B}\boldsymbol{u}$ is a linear transformation of $\boldsymbol{q}$ and $\boldsymbol{u}$ into $\boldsymbol{x}$, then the composition $Q(\boldsymbol{A}\boldsymbol{q} + \boldsymbol{B}\boldsymbol{u}) = \widetilde{Q}(\boldsymbol{q}, \boldsymbol{u})$ is a quadratic function over $\boldsymbol{q}$ and $\boldsymbol{u}$.

3. **Conditioning: Fixing one variable gives a quadratic over the others.** If $Q(\boldsymbol{x}, \boldsymbol{u})$ is a joint quadratic defined over both $\boldsymbol{x}$ and $\boldsymbol{u}$, fixing on of the variables, say $\boldsymbol{x}$ at a particular value $\boldsymbol{x}_t$, gives a quadratic function $Q(\boldsymbol{x}_t, \boldsymbol{u}) = \widetilde{Q}_{\boldsymbol{x}_t}(\boldsymbol{u})$ over the remaining variables $\boldsymbol{u}$.

4. **Marginalization: Optimizing over a subset of variables gives a quadratic over the rest.** Let $Q(\boldsymbol{x}, \boldsymbol{u})$ be a quadratic over both $\boldsymbol{x}$ and $\boldsymbol{u}$. We can solve analytically for the optimal $\boldsymbol{u}$ as a function of $\boldsymbol{x}$ to get an expression $\boldsymbol{u}^*(\boldsymbol{x})$ that tells us the optimal setting to the variable $\boldsymbol{u}$ given the value $\boldsymbol{x}$. Plugging that expression back into the quadratic removes $\boldsymbol{u}$ (since we're optimizing over it) and gives $\widetilde{Q}(\boldsymbol{x}) = Q(\boldsymbol{x}, \boldsymbol{u}^*(\boldsymbol{x}))$. This new function $\widetilde{Q}(\boldsymbol{x})$ is also a quadratic.

It's important to be able to perform the calculations to explicitly derive the above results, but it's more important to understand these results at an intuitive level since their calculation can be tedious and understanding them alone can give a lot of theoretical insight into optimization and control problems. In particular, the last two properties, denoted as *conditioning* and *marginalization* in reference to their corresponding Gaussian analogs (Gaussian inference and quadratic optimization are very closely related), are of central importance to optimal control.