Linear Feedback Control in 3 Easy Steps

Jon Woolfrey*

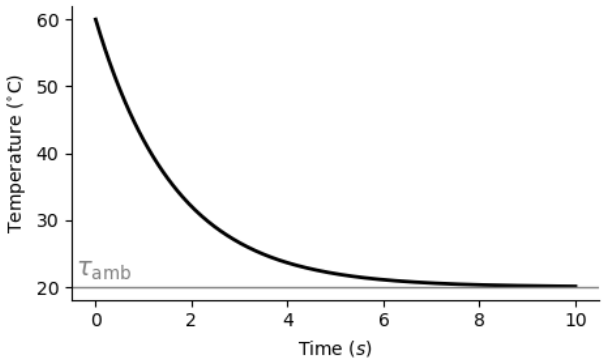
June 10, 2025

Abstract: **In this article I reveal my approach to solving (almost) all control problems. The first is to start from a physical principle. We are all bound by the laws of physics, so using physics principles to formulate control laws leads to elegant solutions. The second is a simple 3-step process to derive a stable, feedback control law. I apply this to first order, and second order systems. At the end I show how it can be applied control a robot arm.**

1. Thinking Like a Physicist

There are many natural phenomena that exhibit stable behaviour. Or at the very least, their dynamic behaviour remains consistent over time. For example, the temperature of a hot cup of coffee will eventually cool until it equalizes around room temperature (Fig. 1).

(a) A cappuccino I made in 2007.



(b) Temperature over time decreases.

Figure 1: The temperate of hot coffee will decay asymptotically toward the ambient temperature.

The rate of change in the temperature of the coffee $\frac{d\tau}{dt} = \dot{\tau}(t)$ (K/s) is proportional to the difference between its current temperature $\tau(t)$ (K) and the ambient temperature $\tau_{amb.}$ (K):

$$\dot{\tau}(t) = k \cdot (\tau_{amb.} - \tau(t)) \tag{1}$$

where $k \in \mathbb{R}^+$ (1/s) is some (positive) constant. This is an ordinary differential equation, so the solution for $\tau(t)$ is an exponential:

$$\tau(t) = e^{-kt}\tau(0) + (1 - e^{-kt})\tau_{amb.} \tag{2}$$

Equation (2) is a little cumbersome. If we instead consider the *difference* in temperature we obtain a much neater equation:

$$\epsilon(t) = \tau_{amb.} - \tau(t) \tag{3a}$$

$$\dot{\epsilon}(t) = -\dot{\tau} \tag{3b}$$

$$= -k \cdot \underbrace{(\tau_{amb.} - \tau(t))}_{\epsilon(t)} \tag{3c}$$

*jonathan.woolfrey@gmail.com

Again, the solution is an exponential:

$$\therefore \epsilon(t) = e^{-kt}\epsilon(0). \tag{4}$$

The *difference* in the temperature will decay toward zero, whether the coffee is hotter or colder than the surrounding environment (Fig. 2).

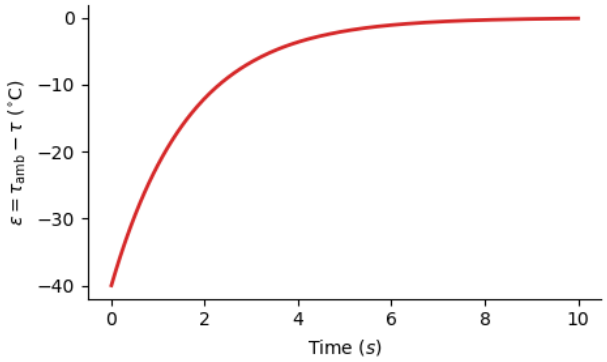


Figure 2: The difference in temperature converges asymptotically toward zero.

All autonomous systems are bound by the laws of physics. By framing control problems with respect to natural laws and observed phenomena, we not only obtain elegant solutions, but equations that are easy to interpret.

2. A 3-Step Process

As with the heat decay of coffee, the objective of control is to force the difference between a desired system state and and the actual system state, i.e. the error, to exponentially decay. This can be solved using a straightforward 3-step process:

1. Define the position error $\boldsymbol{\epsilon} = \mathbf{x}_d - \mathbf{x}$
2. Evaluate how the error evolves with time $\dot{\boldsymbol{\epsilon}} = ?$
3. Design the control input to force the errors to decay: $\dot{\boldsymbol{\epsilon}} = -\mathbf{K}\boldsymbol{\epsilon} \implies \boldsymbol{\epsilon} = e^{-\mathbf{K}t}\boldsymbol{\epsilon}_0$

The most complicated step is the 3rd, as the form of the control equation depends on the structure of the system.

3. First Order Systems

Consider a linear, control affine system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{5}$$

where:

- $\mathbf{x} \in \mathbb{R}^m$ is the configuration,
- $\dot{\mathbf{x}} \in \mathbb{R}^m$ is its time derivative,
- $\mathbf{A} \in \mathbb{R}^{m \times m}$ is the state transition matrix,
- $\mathbf{B} \in \mathbb{R}^{m \times n}$ is the control matrix,
- $\mathbf{u} \in \mathbb{R}^n$ is the control input.

Suppose now we have a desired state $\mathbf{x}_d, \dot{\mathbf{x}}_d \in \mathbb{R}^m$ (typically given by a trajectory), and we want to solve for the control input $\mathbf{u}$ such that $\mathbf{x} \to \mathbf{x}_d$.

Following the 3-step process we:

1. Denote the error from the desired configuration:

$$\boldsymbol{\epsilon} = \mathbf{x}_d - \mathbf{x}. \tag{6}$$

2. Evaluate the time derivative:

$$\begin{aligned} \dot{\boldsymbol{\epsilon}} &= \dot{\mathbf{x}}_d - \dot{\mathbf{x}} && (7a)\\ &= \dot{\mathbf{x}}_d - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}. && (7b) \end{aligned}$$

Now we need to determine the control input $\mathbf{u}$ that forces the error derivative to be (negatively) proportional to itself:

$$\dot{\boldsymbol{\epsilon}} = -\mathbf{K}\boldsymbol{\epsilon} \quad \Longrightarrow \quad \boldsymbol{\epsilon} = e^{-\mathbf{K}t}\boldsymbol{\epsilon}_0 \quad \Longrightarrow \quad \lim_{t\to\infty} \boldsymbol{\epsilon}(t) = \mathbf{0} \tag{8}$$

where $\mathbf{K} \in \mathbb{R}^{m\times m}$ is a gain matrix. This will cause the error to decay to zero over time.

3. Equate the relationship we desire, and solve for the control input $\mathbf{u}$:

$$\begin{aligned} \overbrace{\dot{\mathbf{x}}_d - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u}}^{\dot{\boldsymbol{\epsilon}}} &= -\mathbf{K}\boldsymbol{\varepsilon} && (9a)\\ \mathbf{B}\mathbf{u} &= \dot{\mathbf{x}}_d + \mathbf{K}\boldsymbol{\epsilon} - \mathbf{A}\mathbf{x} && (9b)\\ \mathbf{u} &= \mathbf{B}^{\dagger}\left(\dot{\mathbf{x}}_d + \mathbf{K}\boldsymbol{\epsilon} - \mathbf{A}\mathbf{x}\right) && (9c) \end{aligned}$$

where:

$$\mathbf{B}^{\dagger} = \begin{cases} \left(\mathbf{B}^{\mathsf{T}}\mathbf{B}\right)^{-1}\mathbf{B}^{\mathsf{T}} & \text{for } m > n \\ \mathbf{B}^{-1} & \text{for } m = n \\ \mathbf{B}^{\mathsf{T}}\left(\mathbf{B}\mathbf{B}^{\mathsf{T}}\right)^{-1} & \text{for } m < n \end{cases} \tag{10}$$

is the (pseudo)inverse of $\mathbf{B}$. Take note of the 3 terms in Eqn. (9c):

1. A feedforward term $\dot{\mathbf{x}}_d$,
2. An error feedback term $\mathbf{K}\boldsymbol{\epsilon}$, and
3. Subtracting the natural dynamics working in our favour $-\mathbf{A}\mathbf{x}$.

Equation (3d) is a scalar, so if the exponent is negative, the system will decay (i.e. stable). If it is positive, then it will grow (unstable). However, Eqn. (8) is a matrix exponential. The system is stable if the matrix $\mathbf{K}$ has positive eigenvalues (such that $-\mathbf{K}$ has negative eigenvalues). Why? That's beyond the scope of this article...!

4. Second Order Systems

The dynamics of many systems is naturally expressed & controlled at the acceleration or force level. The process for solving feedback control for second order system is the same.

1. Define the position error:

$$\boldsymbol{\epsilon} = \mathbf{x}_d - \mathbf{x} \tag{11}$$

2. Evaluate the time derivatives:

$$\begin{aligned} \dot{\boldsymbol{\epsilon}} &= \dot{\mathbf{x}}_d - \dot{\mathbf{x}} && (12a)\\ \ddot{\boldsymbol{\epsilon}} &= \ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}. && (12b) \end{aligned}$$

To make the problem more tractable, we can express this second order system as an equivalent first order system and apply the same rules:

$$\begin{bmatrix} \dot{\boldsymbol{\epsilon}} \\ \ddot{\boldsymbol{\epsilon}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ ? & ? \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix}. \tag{13}$$

That is, we need to make the derivatives proportional to the anti-derivatives.

3. If we choose:

$$\ddot{\boldsymbol{\epsilon}} = -\mathbf{D}\dot{\boldsymbol{\epsilon}} - \mathbf{K}\boldsymbol{\epsilon} \tag{14}$$

where:

- $\mathbf{D} \in \mathbb{R}^{m \times m}$ is a gain on the velocity error, and
- $\mathbf{K} \in \mathbb{R}^{m \times m}$ is a gain on the position error.

Now the system becomes:

$$\begin{bmatrix} \dot{\boldsymbol{\epsilon}} \\ \ddot{\boldsymbol{\epsilon}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\epsilon} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix}. \tag{14}$$

We can choose values for $\mathbf{K}$ and $\mathbf{D}$ such that this composite matrix has negative eigenvalues.

By equating Eqn. (12b) and (14) we can reverse engineer the "input" $\ddot{\mathbf{x}}$ as:

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_d + \mathbf{D}\dot{\boldsymbol{\epsilon}} + \mathbf{K}\boldsymbol{\epsilon}. \tag{15}$$

Note the contribution of the 3 terms to the overall control equation:

1. Feedforward acceleration $\ddot{\mathbf{x}}_d$,
2. Velocity error feedback $\mathbf{D}\dot{\boldsymbol{\epsilon}}$, and
3. Position error feedback $\mathbf{K}\boldsymbol{\epsilon}$.

5. A Real Control Problem?

With some abuse of notation, I am going to denote the position & orientation of the endpoint of a robot arm as:

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) : \mathbb{R}^n \mapsto \mathbb{R}^m. \tag{16}$$

This is known as forward kinematics. Given some joint configuration $\mathbf{q} \in \mathbb{R}^n$ we can compute the endpoint pose $\mathbf{x} \in \mathbb{R}^m$ (Fig. 3).

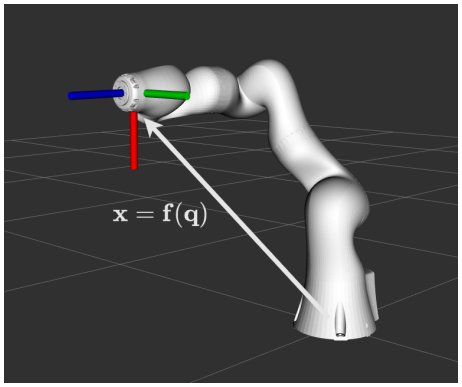


Figure 3: Forward kinematics gives the endpoint pose of a serial link chain given the joint configuration.

We want the endpoint pose to converge on some desired value $\mathbf{x}_d$. To do this we follow the same 3 step process.

1. Define the error:

$$\boldsymbol{\epsilon} = \mathbf{x}_d - \mathbf{f}(\mathbf{q}). \tag{17}$$

2. Differentiate with respect to time:

$$\dot{\boldsymbol{\epsilon}} = \dot{\mathbf{x}}_d - \underbrace{(\partial \mathbf{f}/\partial \mathbf{q})}_{\mathbf{J}(\mathbf{q})} \dot{\mathbf{q}}. \tag{18}$$

The Jacobian $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m \times n}$ can be evaluated numerically from the forward kinematics [Whitney, 1972].

3. Equate the error and solve for the joint velocity $\dot{\mathbf{q}}$:

$$\overbrace{\dot{\mathbf{x}}_d - \mathbf{J}\dot{\mathbf{q}}}^{\dot{\boldsymbol{\epsilon}}} = -\mathbf{K}\boldsymbol{\epsilon} \tag{19a}$$

$$\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}_d + \mathbf{K}\boldsymbol{\epsilon} \tag{19b}$$

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\left(\dot{\mathbf{x}}_d + \mathbf{K}\boldsymbol{\epsilon}\right) \tag{19c}$$

where $\mathbf{J}^{\dagger} \in \mathbb{R}^{n \times m}$ is the (pseudo)inverse, as in Eqn. (10). We can use this method to make the robot follow a trajectory in Cartesian space (Fig. 4). This idea was first proposed in [Whitney, 1969].

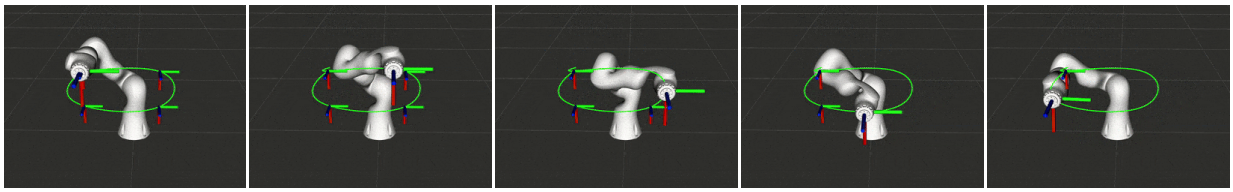


Figure 4: We can follow a trajectory with the endpoint by inverting the Jacobian and solving for the joint velocities.

References

[Whitney, 1969] Whitney, D. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, pages 47–53.

[Whitney, 1972] Whitney, D. (1972). The mathematics of coordinated control of prosthetic arms and manipulators. *Journal of Dynamic Systems, Measurement, and Control*, pages 303–309.