

7

Control

7.1 Motor model with feedforward control	83
7.2 Simple feedback control	85
7.3 Sensor delays	87
7.4 Inertia	90

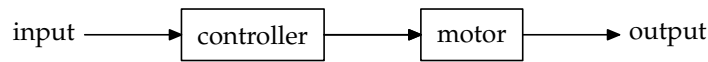
The goals of this chapter are to study:

- how to use feedback to control a system;
- how slow sensors destabilize a feedback system; and
- how to model inertia and how it destabilizes a feedback system.

A common engineering design problem is to control a system that integrates. For example, position a rod attached to a motor that turns input (control) voltage into angular velocity. The goal is an angle whereas the control variable, angular velocity, is one derivative different from angle. We first make a discrete-time model of such a system and try to control it without feedback. To solve the problems of the feedforward setup, we then introduce feedback and analyze its effects.

7.1 Motor model with feedforward control

We would like to design a controller that tells the motor how to place the arm at a given position. The simplest controller is entirely feedforward in that it does not use information about the actual angle. Then the high-level block diagram of the controller–motor system is



where we have to figure out what the output and input signals represent.

A useful input signal is the desired angle of the arm. This angle may vary with time, as it would for a robot arm being directed toward a teacup (for a robot that enjoys teatime).

The output signal should be the variable that interests us: the position (angle) of the arm. That choice helps later when we analyze feedback controllers, which use the output signal to decide what to tell the motor. With the output signal being the same kind of quantity as the input signal (both are angles), a feedback controller can easily compute the error signal by subtracting the output from the input.

With this setup, the controller–motor system takes the desired angle as its input signal and produces the actual angle of the arm as its output.

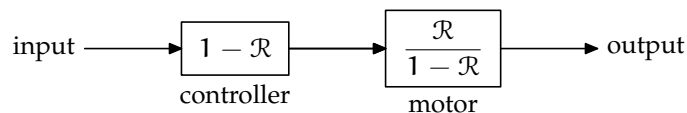
To design the controller, we need to model the motor. The motor turns a voltage into the arm’s angular velocity ω . The continuous-time system that turns ω into angle is $\theta \propto \int \omega \, dt$. Its forward-Euler approximation is the difference equation

$$y[n] = y[n - 1] + x[n - 1].$$

The corresponding system functional is $\mathcal{R}/(1 - \mathcal{R})$, which represents an accumulator with a delay.

Exercise 37. Draw the corresponding block diagram.

The ideal output signal would be a copy of the input signal, and the corresponding system functional would be 1. Since the motor’s system functional is $\mathcal{R}/(1 - \mathcal{R})$, the controller’s should be $(1 - \mathcal{R})/\mathcal{R}$. Sadly, time travel is not (yet?) available, so a bare $\mathcal{R}$ in a denominator, which represents a negative delay, is impossible. A realizable controller is $1 - \mathcal{R}$, which produces a single delay $\mathcal{R}$ for the combined system functional:



Alas, the $1 - \mathcal{R}$ controller is sensitive to the particulars of the motor and of our model of it. Suppose that the arm starts with a non-zero angle before the motor turns on (for example, the whole system gets rotated without the motor knowing about it). Then the output angle remains incorrect by this initial angle. This situation is dangerous if the arm belongs to a 1500-kg robot where an error of 10° means that its arm crashes through a brick wall rather than stopping to pick up the teacup near the wall.

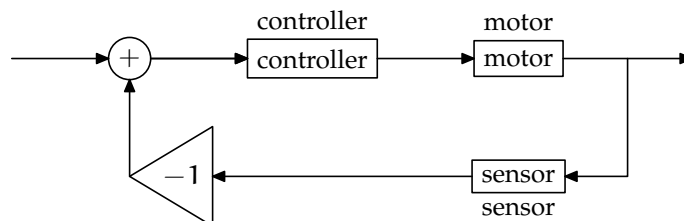
A problem in the same category is an error in the constant of proportionality. Suppose that the motor model underestimates the conversion between voltage and angular velocity, say by a factor of 1.5. Then the system functional of the controller–motor system is $1.5\mathcal{R}$ rather than $\mathcal{R}$. A 500-kg arm might again arrive at the far side of a brick wall.

One remedy for these problems is feedback control, whose analysis is the subject of the next sections.

7.2 Simple feedback control

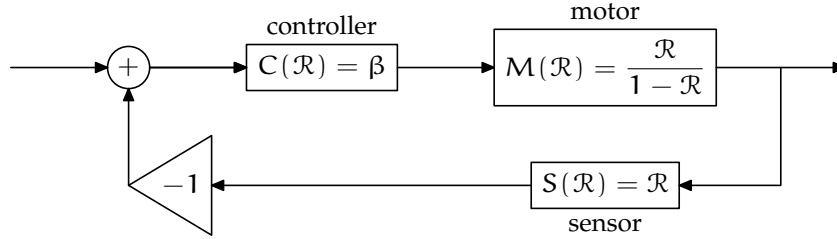
In feedback control, the controller uses the output signal to decide what to tell the motor. Knowing the input and output signals, an infinitely intelligent controller could deduce how the motor works. Such a controller would realize that the arm's angle starts with an offset or that the motor's conversion is incorrect by a factor of 1.5, and it would compensate for those and other problems. That mythical controller is beyond the scope of this course (and maybe of all courses). In this course, we use only linear-systems theory rather than strong AI. But the essential and transferable idea in the mythical controller is feedback.

So, sense the the angle of the arm, compare it to the desired angle, and use the difference (the error signal) to decide the motor's speed:



A real sensor cannot respond instantaneously, so assume the next-best situation, that the sensor includes one unit of delay. Then the sensor's output gets subtracted from the desired angle to get the error signal, which is used

by the controller. The simplest controller, which uses so-called proportional control, just multiplies the error signal by a constant β . This setup has the block diagram



It was analyzed in lecture and has the system functional

$$\frac{C(\mathcal{R})M(\mathcal{R})}{1 + C(\mathcal{R})M(\mathcal{R})S(\mathcal{R})} = \frac{\beta\mathcal{R}/(1 - \mathcal{R})}{1 + \beta\mathcal{R}^2/(1 - \mathcal{R})}.$$

Multiply by $(1 - \mathcal{R})/(1 - \mathcal{R})$ to clear the fractions. Then

$$F(\mathcal{R}) = \frac{\beta\mathcal{R}}{1 - \mathcal{R} + \beta\mathcal{R}^2},$$

where $F(\mathcal{R})$ is the functional for the whole feedback system.

Let's analyze its behavior in the extreme cases of the gain β . As $\beta \rightarrow \infty$, the system functional limits to $\mathcal{R}/\mathcal{R}^2 = 1/\mathcal{R}$, which is a time machine. Since we cannot build a time machine just by choosing a huge gain in a feedback system, some effect should prevent us raising $\beta \rightarrow \infty$. Indeed, instability will prevent it, as we will see by smoothly raising β from 0 to ∞ .

To study stability, look at the poles of the feedback system, which are given by the factors of the denominator $1 - \mathcal{R} + \beta\mathcal{R}^2$. The factored form is $(1 - p_1\mathcal{R})(1 - p_2\mathcal{R})$. So the sum of the poles is 1 and their product is β . At the $\beta \rightarrow 0$ extreme, which means no feedback, the poles are approximately $1 - \beta$ and β . The pole near 1 means that the system is almost an accumulator; it approximately integrates the input signal. This behavior is what the motor does without feedback and is far from our goal that the controller-motor system copy the input to the output.

Turning up the gain improves the control. However, at the $\beta \rightarrow \infty$ extreme, the poles are roughly at

$$p_{1,2} \approx \frac{1}{2} \pm j\sqrt{\beta}$$

so that $p_1 p_2 = \beta$. Those poles lie far outside the unit circle, making the system highly unstable. *Too much gain destabilizes a feedback system.*

To find the best gain, study the poles. The pole farthest from the origin has the most rapidly growing, or most slowly decaying output. Therefore, to make the system as stable as possible, minimize the distance to the pole most distant from the origin. To do so, place poles at the same location. In this example, the location must be $p_1 = p_2 = 1/2$ since $p_1 + p_2 = 1$. Since $\beta = p_1 p_2$, choose $\beta = 1/4$. Now the output position rapidly approaches the desired position. Equivalently, in response to an impulse input signal, the error signal decays rapidly to zero, roughly halving each time step.

Exercise 38. Why does the error signal *roughly* halve, rather than exactly halve with every time step?

7.3 Sensor delays

The preceding model contained a rapid sensor. Suppose instead that the sensor is slow, say $S(\mathcal{R}) = \mathcal{R}^2$.

Pause to try 34. With this sensor, what is the functional for the feedback system?

The functional for the feedback system is

$$\frac{\beta \mathcal{R}}{1 - \mathcal{R} + \beta \mathcal{R}^3},$$

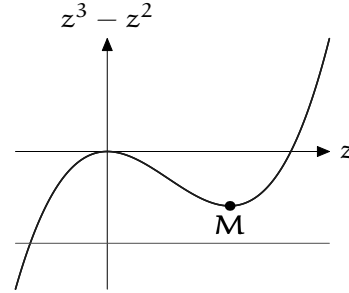
which is the previous functional with the $\mathcal{R}^2$ in the denominator replaced by $\mathcal{R}^3$ because of the extra power of $\mathcal{R}$ in the sensor functional. There are many analyses that one can do on this system. For simplicity, we choose a particular gain β – the rapid-convergence gain with the fast sensor – and see how the extra sensor delay moves the poles. But before analyzing, predict the conclusion!

Pause to try 35. Will the extra sensor delay move the least stable pole inward, outward, or leave its magnitude unchanged?

The poles are at the roots of the corresponding equation $z^3 - z^2 + \beta = 0$ or

$$z^3 - z^2 = -\beta.$$

Here is a sketch of the curve $z^3 - z^2$. Its minimum is at $M = (2/3, -4/27)$, so the horizontal line at $-1/4$ intersects the curve only once, in the left half of the plane. The equation therefore has one (negative) real root and two complex roots. So, for $\beta = 1/4$, the system has two complex poles and one real pole. The following Python code finds the poles. It first finds the real pole p_1 using the Newton–Raphson [18] method of successive approximation. The Newton–Raphson code is available as part of the `scipy` package. The real pole constrains the real part of the complex poles because the sum of the poles $p_1 + p_2 + p_3$ is 1, and the two complex poles have the same real part. So



$$\operatorname{Re} p_{2,3} = \frac{1 - p_1}{2}.$$

To find the imaginary part of p_2 or p_3 , use the product of the poles. The product $p_1 p_2 p_3$ is $-\beta$. Since the magnitudes of the complex poles are equal because they are complex conjugates, we have

$$|p_{2,3}| = \frac{\sqrt{-\beta}}{p_1}.$$

Then find the imaginary part of one complex pole from the computed real part and magnitude. This algorithm is implemented below.

```
from scipy import *

def poly(beta):
    def f(z): # closure that knows the passed-in value of
    beta
```

```

        return z**3-z**2+beta
    return f

# return the three poles of 1-R+beta*R^3 by solving z^3-z^2+beta=0
# method works for beta>4/27 (one real root, two complex roots)
def solve(beta):
    # use Newton-Raphson to find the one real root (for beta>4/27)
    realroot = optimize.newton(poly(beta), 1.0)
    # use realroot to find complex roots
    realpart = (1-realroot)/2          # sum of the roots
    is 1
    magnitude = sqrt(-beta/realroot)   # product of roots is
    -beta
    imaginarypart = sqrt(magnitude**2 - realpart**2)
    complexroot = realpart + 1j*imaginarypart
    return (realroot, complexroot, conjugate(complexroot))

```

The result is

$$\begin{aligned}
 p_1 &\approx -0.419, \\
 p_2 &\approx 0.710 + 0.303j, \\
 p_3 &\approx 0.710 - 0.303j.
 \end{aligned}$$

With these locations, the complex poles are the least stable modes. These poles have a magnitude of approximately 0.772. In the previous system with the fast sensor and the same gain $\beta = 1/4$, both poles had magnitude 0.5. So the sensor delay has made the system more unstable and, since the poles are complex, has introduced oscillations.

To make the system more stable, one can reduce β . But this method has problems. The $\beta \rightarrow \infty$ limit makes the feedback system turn into the system $\mathcal{R}^{-2}$, independent of the motor's characteristics. The other direction, reducing β , exposes more particulars of the motor, making a feedback system sensitive to the parameters of the motor. Thus lower β means one gives up some advantages of feedback. No choices are easy if the sensor delay is long. When β is small, the system system is stable but benefits hardly at all from feedback.

Pause to try 36. What happens when β is large?

When β is large, then the feedback system is less stable and eventually unstable. To prove this, look at how the denominator $1 - \mathcal{R} + \beta\mathcal{R}^3$ constrains the location of the poles. The product of the poles is the negative of the coefficient of $\mathcal{R}^3$, so $p_1 p_2 p_3 = -\beta$. Using magnitudes,

$$|p_1| |p_2| |p_3| = \beta,$$

so when $\beta > 1$, at least one pole must have magnitude greater than one, meaning that it lies outside the unit circle.

From the analysis with $S(\mathcal{R}) = \mathcal{R}$ and $S(\mathcal{R}) = \mathcal{R}^2$, try to guess what happens with one more delay in the sensor, which makes $S(\mathcal{R}) = \mathcal{R}^3$ (again with $\beta = 1/4$).

Exercise 39.

What happens to the stability if the sensor has yet another delay, so $S(\mathcal{R}) = \mathcal{R}^3$? First guess then check your guess by finding the poles numerically or otherwise.

7.4 Inertia

Now return to the quick sensor with one delay, but improve the physical model of the motor. A physical motor cannot change its angular velocity arbitrarily quickly, especially with a heavy rod attached to it. Instant changes imply zero inertia. So we should add inertia to the model of the motor.

A simple model of inertia is a new term in the difference equation:

$$y[n] = y[n-1] + x[n-1] + \underbrace{\frac{1}{2}(y[n-1] - y[n-2])}_{\text{inertia}}.$$

The difference $y[n-1] - y[n-2]$ estimates the motor's angular velocity. The coefficient of $1/2$ means that, with each time step, the motor gets rid of one-half of its previous angular velocity. Alternatively, it means that one-half of its previous angular velocity remains to affect the new angle. This coefficient depends on the mass of the motor and the mass and length of the rod – more exactly, on the moment of inertia of the system – and on the

power of the motor. For illustrating the ideas, we picked the convenient coefficient of $1/2$.

The motor's system functional, which was $\mathcal{R}/(1 - \mathcal{R})$, becomes

$$M(\mathcal{R}) = \frac{\mathcal{R}}{1 - \frac{3}{2}\mathcal{R} + \frac{1}{2}\mathcal{R}^2}.$$

Pause to try 37. Find the poles of this system and mark them on a pole-zero diagram.

This functional factors into

$$M(\mathcal{R}) = \frac{\mathcal{R}}{(1 - \frac{1}{2}\mathcal{R})(1 - \mathcal{R})}$$

so the poles are $p_1 = 1/2$ and $p_2 = 1$.

The functional for the feedback system of controller $C(\mathcal{R})$, sensor $S(\mathcal{R})$, and motor $M(\mathcal{R})$ is

$$F(\mathcal{R}) = \frac{C(\mathcal{R})M(\mathcal{R})}{1 + C(\mathcal{R})M(\mathcal{R})S(\mathcal{R})},$$

With the usual controller $C(\mathcal{R}) = \beta$, fast sensor $S(\mathcal{R}) = \mathcal{R}$, and new motor model $M(\mathcal{R})$ with inertia, the feedback system has the functional

$$\frac{\beta\mathcal{R}/(1 - \frac{3}{2}\mathcal{R} + \frac{1}{2}\mathcal{R}^2)}{1 + \beta\mathcal{R}^2/(1 - \frac{3}{2}\mathcal{R} + \frac{1}{2}\mathcal{R}^2)}.$$

Clear the fractions to get

$$F(\mathcal{R}) = \frac{\beta\mathcal{R}}{1 - \frac{3}{2}\mathcal{R} + (\beta + \frac{1}{2})\mathcal{R}^2}.$$

This denominator is quadratic so we can find the poles for all β without needing numerical solutions. So let β increase from 0 to ∞ . Their locations are determined by factoring the denominator. When $\beta = 0$, it factors into $(1 - \mathcal{R}/2)(1 - \mathcal{R})$, and the poles are at $1/2$ and 1 – which are the poles of the motor itself. The pole at 1 indicates an accumulator, which means that the system is very different than one that copies the input signal to the output

signal. But we knew it would happen that way, because choosing $\beta = 0$ turns off feedback.

As β increases, the poles move. The sum $p_1 + p_2$ remains constant at $3/2$, so the poles are at $3/4 \pm \alpha$. For $\beta = 0$, the α is $1/4$. As β increases, α increases and the poles slide along the real axis until they collide at $p_{1,2} = 3/4$. When they collide, the product of the poles is $p_1 p_2 = 9/16$. This product is the coefficient of $\mathcal{R}^2$, which is $1/2 + \beta$. So $1/2 + \beta = 9/16$, which means that the poles collide when $\beta = 1/16$. That controller gain results in the most stable system. It is also significantly smaller than the corresponding gain when the motor had no inertia. This simple controller that only has a gain has difficulty compensating for inertia.

Pause to try 38. For what β do the poles cross the unit circle into instability? Compare that critical β with the corresponding value in the model without inertia.

As β increases farther, the poles move along a vertical line with real part $3/4$. The next interesting β is when the poles hit the unit circle. Their product is then 1, which is the coefficient of $\mathcal{R}^2$ in the denominator of the system functional. So $1/2 + \beta = 1$ or $\beta = 1/2$. The resulting poles are

$$p_{1,2} = \frac{3}{4} \pm j\frac{\sqrt{7}}{4}.$$

In the model without inertia, β could increase to 1 before the feedback system became unstable, whereas now it can increase only till $1/2$: *Inertia destabilizes the feedback system.*

Exercise 40. Sketch how the poles move as β changes from 0 to ∞ .

Exercise 41.

What if the system has more inertia, meaning that old angular velocities persist longer? For example:

$$y[n] = y[n-1] + x[n-1] + \underbrace{\frac{4}{5}(y[n-1] - y[n-2])}_{\text{inertia}}.$$

Sketch how the poles of the feedback system move as β changes from 0 to ∞ , and compare with the case of no inertia and of inertia with a coefficient of $1/2$.

MIT OpenCourseWare
<https://ocw.mit.edu/>

RES.6-011 Discrete-Time Signals and Systems: An Operator Approach
Spring 2026

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.