

## Chapter 2 Continuous Dynamics

1. A **tuning fork**, shown in Figure 2.1, consists of a metal finger (called a **tine**) that is displaced by striking it with a hammer. After being displaced, it vibrates. If the tine has no friction, it will vibrate forever. We can denote the displacement of the tine after being struck at time zero as a function  $y: \mathbb{R}_+ \rightarrow \mathbb{R}$ . If we assume that the initial displacement introduced by the hammer is one unit, then using our knowledge of physics we can determine that for all  $t \in \mathbb{R}_+$ , the displacement satisfies the differential equation

$$\ddot{y}(t) = -\omega_0^2 y(t)$$

where  $\omega_0^2$  is a constant that depends on the mass and stiffness of the tine, and where  $\ddot{y}(t)$  denotes the second derivative with respect to time of  $y$ . It is easy to verify that  $y$  given by

$$\forall t \in \mathbb{R}_+, \quad y(t) = \cos(\omega_0 t)$$

is a solution to the differential equation (just take its second derivative). Thus, the displacement of the tuning fork is sinusoidal. If we choose materials for the tuning fork so that  $\omega_0 = 2\pi \times 440$  radians/second, then the tuning fork will produce the tone of A-440 on the musical scale.

- (a) Is  $y(t) = \cos(\omega_0 t)$  the only solution? If not, give some others.

**Solution:** The following is a solution for any constant  $\alpha$ :

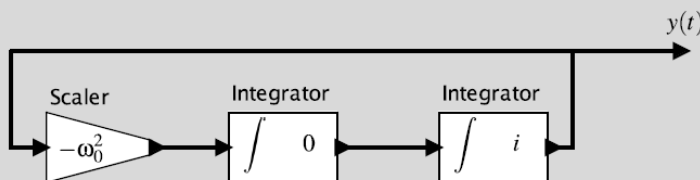
$$y(t) = \alpha \cos(\omega_0 t) .$$

- (b) Assuming the solution is  $y(t) = \cos(\omega_0 t)$ , what is the initial displacement?

**Solution:**  $y(0) = \cos(\omega_0 \times 0) = 1$ .

- (c) Construct a model of the tuning fork that produces  $y$  as an output using generic actors like Integrator, adder, scaler, or similarly simple actors. Treat the initial displacement as a parameter. Carefully label your diagram.

**Solution:** The following model will do the job:



Here,  $i$  is the initial displacement.

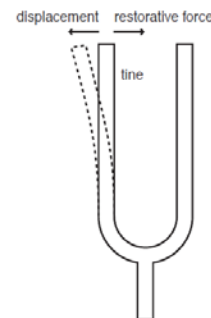


Figure 2.1: A tuning fork.

3. This exercise studies linearity.

- (a) Show that the helicopter model defined in Example 2.1 is linear if and only if the initial angular velocity  $\dot{\theta}_y(0) = 0$ .

**Solution:** The input  $T_y$  and output  $\dot{\theta}_y$  are related by

$$\dot{\theta}_y(t) = \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau.$$

First, we need to show that if  $\dot{\theta}_y(0) = 0$ , then superposition applies. Then we need to show if  $\dot{\theta}_y(0) \neq 0$ , superposition does not apply. For the first problem, if  $\dot{\theta}_y(0) = 0$  then we have

$$\dot{\theta}_y(t) = \frac{1}{I_{yy}} \int_0^t T_y(\tau) d\tau.$$

Suppose the input is given by

$$T_y = aT_1 + bT_2$$

where  $a$  and  $b$  are real numbers and  $T_1$  and  $T_2$  are signals. Then the output is

$$\begin{aligned} \dot{\theta}_y(t) &= \frac{1}{I_{yy}} \int_0^t (aT_1(\tau) + bT_2(\tau)) d\tau \\ &= \frac{a}{I_{yy}} \int_0^t T_1(\tau) d\tau + \frac{b}{I_{yy}} \int_0^t T_2(\tau) d\tau. \end{aligned}$$

It is easy to see that the first term is  $a$  times what the output would be if the input were only  $T_1$ , and the second term is  $b$  times what the output would be if the input were only  $T_2$ . That is, if the system function is  $S$ , the output is

$$\dot{\theta}_y(t) = a(S(T_1))(t) + b(S(T_2))(t).$$

Next, assume that  $\dot{\theta}_y(0) \neq 0$ . With the same input as above, we get the output

$$\begin{aligned} \dot{\theta}_y(t) &= \dot{\theta}_y(0) + \frac{1}{I_{yy}} \int_0^t (aT_1(\tau) + bT_2(\tau)) d\tau \\ &= \dot{\theta}_y(0) + \frac{a}{I_{yy}} \int_0^t T_1(\tau) d\tau + \frac{b}{I_{yy}} \int_0^t T_2(\tau) d\tau. \end{aligned}$$

We can now see that the output is

$$\dot{\theta}_y(t) = a(S(T_1))(t) + b(S(T_2))(t) - \dot{\theta}_y(0),$$

so superposition does not apply.

(b) Show that the cascade of any two linear actors is linear.

**Solution:** Given an actor with function  $S_1$  and another with function  $S_2$ , the cascade composition is an actor with function  $S_1 \circ S_2$ , the composition of the two functions. If  $S_1$  and  $S_2$  both satisfy the superposition property, then

$$\begin{aligned} S_2(S_1(ax_1 + bx_2)) &= S_2(aS_1(x_1) + bS_1(x_2)) \\ &= aS_2(S_1(x_1)) + bS_2(S_1(x_2)). \end{aligned}$$

Hence, the composition also satisfies superposition.

(c) Augment the definition of linearity so that it applies to actors with two input signals and one output signal. Show that the adder actor is linear.

**Solution:** A system model  $S: X_1 \times X_2 \rightarrow Y$ , where  $X_1$ ,  $X_2$ , and  $Y$  are sets of signals, is linear if it satisfies the superposition property:

$$\begin{aligned} \forall x_1, x'_1 \in X_1 \text{ and } \forall x_2, x'_2 \in X_2 \text{ and } \forall a, b \in \mathbb{R}, \\ S(ax_1 + bx'_1, ax_2 + bx'_2) = aS(x_1, x_2) + bS(x'_1, x'_2). \end{aligned}$$

The adder component is given by

$$S(x_1, x_2) = x_1 + x_2.$$

Hence

$$\begin{aligned} S(ax_1 + bx'_1, ax_2 + bx'_2) &= ax_1 + bx'_1 + ax_2 + bx'_2 \\ &= a(x_1 + x_2) + b(x'_1 + x'_2) \\ &= aS(x_1, x_2) + bS(x'_1, x'_2). \end{aligned}$$

4. Consider the helicopter of Example 2.1, but with a slightly different definition of the input and output. Suppose that, as in the example, the input is  $T_y: \mathbb{R} \rightarrow \mathbb{R}$ , as in the example, but the output is the position of the tail relative to the main rotor shaft. Specifically, let the  $x$ - $y$  plane be the plane orthogonal to the rotor shaft, and let the position of the tail at time  $t$  be given by a tuple  $((x(t), y(t)))$ . Is this model LTI? Is it BIBO stable?

**Solution:** In this case, the system can be modeled as a function with two output signals,

$$S: (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow (\mathbb{R} \rightarrow \mathbb{R})^2,$$

where

$$(S(T_y))(t) = (x(t), y(t)),$$

where  $(x(t), y(t))$  is the position of the tail in the  $x$ - $y$  plane. This model is clearly not linear. If the input torque doubles, for example, the output values will not double. In fact, the output values are constrained to lie on a circle centered at the origin, regardless of the input. For this reason, the model is BIBO stable. The output is always bounded. Thus, while our previous model was linear and unstable, this one is nonlinear and stable. Which model is more useful?

6. A DC motor produces a torque that is proportional to the current through the windings of the motor. Neglecting friction, the net torque on the motor, therefore, is this torque minus the torque applied by whatever load is connected to the motor. Newton's second law (the rotational version) gives

$$k_T i(t) - x(t) = I \frac{d}{dt} \omega(t), \quad (2.1)$$

where  $k_T$  is the motor torque constant,  $i(t)$  is the current at time  $t$ ,  $x(t)$  is the torque applied by the load at time  $t$ ,  $I$  is the moment of inertia of the motor, and  $\omega(t)$  is the angular velocity of the motor.

- (a) Assuming the motor is initially at rest, rewrite (2.1) as an integral equation.

**Solution:** Integrating both sides, we get

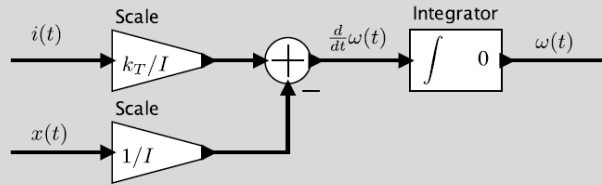
$$\int_0^t (k_T i(\tau) - x(\tau)) d\tau = I \omega(t).$$

Solving for  $\omega(t)$  we get

$$\omega(t) = \frac{1}{I} \int_0^t (k_T i(\tau) - x(\tau)) d\tau.$$

- (b) Assuming that both  $x$  and  $i$  are inputs and  $\omega$  is an output, construct an actor model (a block diagram) that models this motor. You should use only primitive actors such as integrators and basic arithmetic actors such as scale and adder.

**Solution:** A solution is shown below:



- (c) In reality, the input to a DC motor is not a current, but is rather a voltage. If we assume that the inductance of the motor windings is negligible, then the relationship between voltage and current is given by

$$v(t) = Ri(t) + k_b \omega(t),$$

where  $R$  is the resistance of the motor windings and  $k_b$  is a constant called the motor back electromagnetic force constant. The second term appears because a rotating motor also functions as an electrical generator, where the voltage generated is proportional to the angular velocity.

Modify your actor model so that the inputs are  $v$  and  $x$  rather than  $i$  and  $x$ .

**Solution:** A solution is shown below:

