sim.fluid   Nonlinear fluid system example

1   This example gets one started on the design problem Exercise sim..

2   Consider a fluid system with an input volumeric flowrate Q_s into a capacitance C that is drained by only a single pipe of nonlinear resistance R and L, as shown in the linear graph of Figure fluid.1. The nonlinearity of R is a good way to model an overflow. In this lecture, we will derive a nonlinear state-space model for the system—specifically, a state equation—and solve it, numerically using Matlab.

nonlinear state−space model

Normal tree, order, and variables

3   Fig. fluid.1 already shows the normal tree. There are two independent energy storage elements, making it a second-order ($n = 2$) system. We define the state vector to be

$$\boldsymbol{x} = \begin{bmatrix} P_C & Q_L \end{bmatrix}^\top. \qquad (1)$$

The input vector is defined as $\boldsymbol{u} = \begin{bmatrix} Q_s \end{bmatrix}$.

Elemental, continuity, and compatibility equations

4   Before turning to our familiar elemental equations, we'll consider the nonlinear resistor.

Nonlinear elemental equation

5   Suppose we are trying to model an overflow with the pipe R–L to ground. An overflow
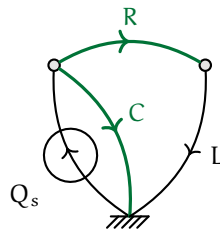

Figure fluid.1: a linear graph and normal tree (green) for a nonlinear fluid system.

would have no flow until the fluid capcitor fills to a certain height, then it would transition to flowing quite rapidly. This process seems to be inherently nonlinear because we cannot write an element that depends linearly on the height of the fluid in the capacitor (even if height was one of our state variables, which it is not).

6   The volume in the tank can be found by integrating in flow (Q_s) minus out flow (Q_R), but this is not accessible within a simulation, since it must be integrated, so it's not an ideal variable for our model. However, the pressure P_C—a state variable—is proportional to the fluid height in the capacitor, which we'll call h:

$$P_C = \rho g h, \tag{2}$$

where ρ is the density of the fluid and g is the gravitational acceleration. Since the height of the capacitor is presumably known, we can use P_C to be our fluid height metric.

7   When the height h reaches a certain level, probably near the capacitor's max, which we'll denote h_m, we want our overflow pipe R-L to start flowing. Since P_C is our height metric, we want to define a resistance as a function of it, $R(P_C)$.

8   Now we must determine the form of $R(P_C)$. Clearly, when $h \sim P_C$ is small, we want as little as possible flow through R-L, so $R(P_C)$ should be large. If R was infinitely large, divisions by zero would likely arise in a simulation, so we choose to set our low-pressure R to some finite value:

$$R(P_C)|_{P_C \to 0} = R_0. \tag{3}$$

Conversely, when $h \sim P_C$ is large (near max), we want maximum flow through R-L, so $R(P_C)$ should be some finite value, say, that of the pipe:

$$R(P_C)|_{P_C \to \infty} = R_\infty. \tag{4}$$

Clearly, this model requires $R_\infty \ll R_0$.

9   The transition from R_0 to R_∞ should be smooth in order to minimize numerical solver difficulties. Furthermore, a smooth transition is consistent with, say, a float opening a valve at the bottom of the capacitor,[2] since the valve would transition continuously from closed to open. Many functions could be used to model this transition, especially if piecewise functions are considered. However, the $\mathtt{tanh}$ function has the merit of enabling us to easily define a single non-piecewise function for the entire domain. Let $\overline{P}_C$ be the transition pressure and ΔP_C be the transition width. A convenient nonlinear resistor, then, is

$$R(P_C) = R_\infty + \frac{R_0 - R_\infty}{2} \left(1 - \tanh \frac{5(P_C - \overline{P}_C)}{\Delta P_C}\right).$$

(5)

Note that this function only approximately satisfies $R(P_C)|_{P_C \to 0} = R_0$, but the small deviation from this constraint is worth it for the convenience it provides. Another noteworthy aspect of Equation 5 is the factor of 5, which arises from the $\mathtt{tanh}$ function's natural transition width, which we alter via ΔP_C.

Other elemental equations and the continuity and compatibility equations

10   The other elemental equations have been previously encountered and are listed in the table, below. Furthermore, continuity and compatibility equations can be found in the usual way—by drawing contours and temporarily creating loops by including links in the normal tree. We proceed by drawing a table of all elements and writing an elemental equation for each element, a continuity equation for each branch of the normal tree, and a compatibility equation for each link.

2. Note that this model might be said to assume the overflow pipe is attached to the bottom of the capacitor since the pressure driving fluid through this pipe is supposed to be P_C. However, no matter the overflow valve's inlet height, if its outlet is at the height of the bottom of the capacitor, this model is still valid.

el.	elemental eq.
C	$\dfrac{dP_C}{dt} = \dfrac{1}{C}Q_C$
L	$\dfrac{dQ_L}{dt} = \dfrac{1}{L}P_L$
R	$P_R = Q_R R(P_C)$

el.	cont/comp. eq.
C	$Q_C = Q_s - Q_L$
L	$P_L = P_C - P_R$
R	$Q_R = Q_L$

State equation

11 The system of equations composed of the elemental, continuity, and compatibility equations can be reduced to the state equation. This equation nonlinear, so it cannot be written in the linear for with A and B matrices. However, it can still be written as a system of first-order ordinary differential equations, as follows:

$$\frac{d\boldsymbol{x}}{dt} = f(\boldsymbol{x}, \boldsymbol{u})$$

$$= \begin{bmatrix} (Q_s - Q_L)/C \\ (P_C - Q_L R(P_C))/L \end{bmatrix}. \qquad (6)$$

Although it appears simple, this nonlinear differential equation likely has no known analytic solution. Two other options are available:

1. linearize the model about an operating point and solve the linearized equation or
2. numerically solve the nonlinear equation.

Both methods are widely useful, but let's assume we require the model to be accurate over

a wide range of capacitor fullness. Therefore, we choose to investigate via numerical solution.

Simulation

Broadly, the numerical investigation will be conducted via Matlab's `ode23t` solver.[3] Of course, as with any numerical solution, specific values of the parameters must be selected. We begin with declaring the fluid to be water, endowing it with a density, and specify the gravitational acceleration `g`. Furthermore, an "anonymous" function `P_fun` is defined, accepting the height `h` of the fluid in the capacitor and returning the corresponding pressure. Other parameters specified include the fluid capacitance `C` and the overflow pipe inertance `L`.

```matlab
global C L % global to be used in state equation
C = 1e3; % ... fluid capacitance
L = 1e-3; % ... fluid inertance
rho = 997; % kg/m^3 ... density of water
g = 9.81; % m/s^2 ... gravitational constant
P_fun = @(h) rho*g*h; % pressure as a function of height
```

Next, we define the maximum height `h_max` of fluid in the capacitor, the transition height `h_t`, and the distance `dh` over which the resistor will transition from high to low impedance.

```matlab
h_max = 1; % m ... maximum height of fluid
h_t = .88; % m ... transition height
dh = .05; % m ... height difference for transition
```

Corresponding pressures, which we prefer for computation, can be computed with `P_fun`.

```matlab
P_t = P_fun(h_t); % N/m^2 ... transition pressure
dP = P_fun(dh); % N/m^2 ... pressure dif for transition
```

Nonlinear resistance

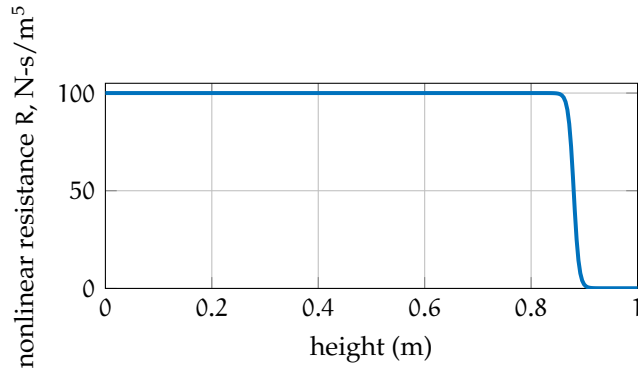Now, let's define the variable resistance function `R_fun` ($R(P_C)$). We define the

3. Source Matlab files can be found on ricopic.one/dynamic_systems_ii/source as ricopic.one/dynamic_systems_ii/source/nonlinear_tank_example.m and ricopic.one/dynamic_systems_ii/source/nonlinear_fluid_state.m.

Figure fluid.2: nonlinear resistance versus tank water level.

anonymous function via the two "limiting" resistances R_0 (R_0) and R_inf (R_∞).

```
R_inf = 1e-1; % N/m^2 / m^3/s ... resistance with full cap
R_0 = 1e2; % ... resistance with empty capacitor
R_fun = @(P) (R_0-R_inf)/2*(1-tanh(5/dP*(P-P_t)))+R_inf;
```

Let's take a moment to plot this function. See Figure fluid.2 for the results. This is a reasonable approximation of a valve that allows no flow until the capacitor fluid height reaches a threshold, then allows a significant amount of flow.

```
h_half = linspace(0,h_t-dh,50);
h_a = [... % heights to plot
  h_half(1:end-1),...
  linspace(h_half(end),h_max,50)...
];
P_a = P_fun(h_a); % N/m^2 ... pressures to plot
```

Numerical solution

The numerical ODE solver we'll use (ode23t) requires we define the first-order system of differential equations from Equation 6. This is done by writing a function file nonlinear_fluid_state.m that the function return the time derivative of the state vector x_a (x) at a given time.

```matlab
function dx = nonlinear_fluid_state(t,x,u_fun,R_fun)
global C L
% x(1) is P_C
% x(2) is Q_L
% R_fun is the nonlinear resistance

% call input function at this time step
Q_s = u_fun(t);

% compute nonlinear resistance at this time step
R = R_fun(x(1));

dx = zeros(2,1);     % a column vector
dx(1) = (Q_s - x(2))/C;     % d P_C/dt
dx(2) = (x(1) - x(2)*R)/L; % d Q_L/dt
end
```

We also pass it the nonlinear resistance function
`R_fun` and the input function `Q_s_fun`. Let's
model an offset sinusoidal input flowrate,
defined as an anonymous function as follows.

```matlab
Q_s_fun = @(t) 1e4*(1+sin(2*pi/1e2*t));
```

We're ready to simulate! The time array and
zero initial conditions are specified, then
simulation commences. There are several
Matlab ODE solver routines with the same or
similar syntax. Many ODEs can be solved with
the `ode45` function. However, this problem is
what is called "stiff," which runs much better on
the solver `ode23t`.

```matlab
t_a = linspace(0,1.4e3,1e3);
x_0 = [0;0];
x_sol_struc = ode23t(...
  @(t,x) nonlinear_fluid_state(t,x,Q_s_fun,R_fun),...
  t_a,...
  x_0...
);
x_sol = deval(x_sol_struc,t_a);
```

We plot the results in Figure fluid.3. So the
overflow is relatively inactive while the
capacitor fills, until P_C achieves the pressure
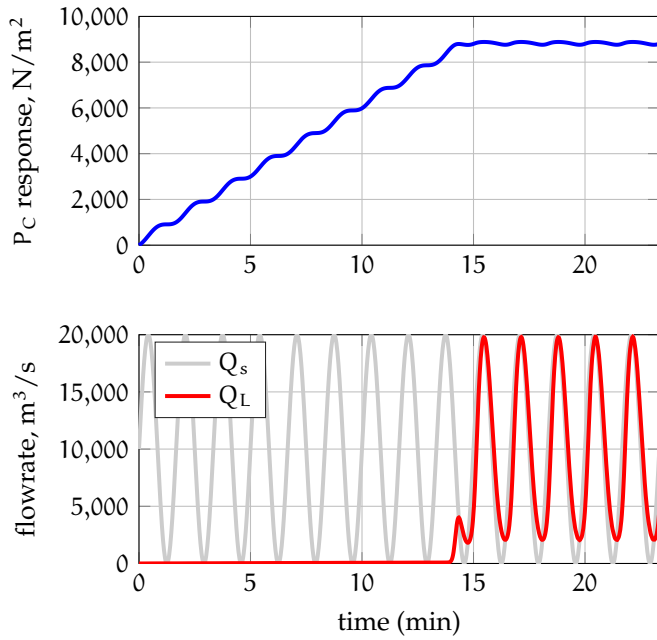associated with a near-full capacitor. Then the

Figure fluid.3: state response to input Q_s.

flowrate suddenly increases rapidly due to the
sudden drop in $R(P_C)$. Since the input is
oscillating, the overflow pipe loses flowrate,
then gains it again when the input flowrate
increases enough to increase the capacitor
pressure.