

```

% Author: Noah J. Cowan
% Date: Oct 29, 2025
% Airbag Inflation Simulation
% Models pump filling airbag with back-pressure effects

% This script is illustrative, and not meant to be a detailed mechanistic
% simulation. The simplifying assumptions are listed below.

clear all; close all;

%% Parameters
% This parameter set represents a simplified buoyancy control system
% for autonomous pool cleaning robots, based on practical considerations.

% Pump characteristics (illustrative):
Q_nominal = 4.0/60;    % Assumed nominal pump flow rate [L/sec]
P_max = 50;           % Maximum pressure of pump [kPa]

% Airbag characteristics (illustrative):
V_slack = 1;          % Slack (rest) volume [L] – membrane just taut above this
V_scale = 1;          % Characteristic stretch volume [L] (sets the "knee")
k = 10.0;             % Nonlinear coefficient [kPa]

% Design parameters (illustrative)
V_thresh = 1.75;      % Minimum gas volume threshold for system buoyancy
                    % Assumption: V > V_thresh ==> the robot floats.
t_predetermined = 40; % Assume a predetermined time of 40 seconds. Time
                    % selected to achieve a predetermined final volume
                    % that exceeds the volume threshold i.e.
                    % V_predetermined > V_thresh

% Straightforward calculation of the asymptotic volume
V_asymptote = V_slack + V_scale * log(1 + P_max/k);

% Simulation parameters for simple Euler integration
dt = 0.01;            % Time step [sec]
t_max = 80;           % Maximum simulation time [sec]
t = 0:dt:t_max;       % Time vector

%% Initialize arrays
V = zeros(size(t));    % Volume [L]
P = zeros(size(t));    % Pressure [kPa]
Q = zeros(size(t));    % Actual flow rate [L/sec]

% Initial conditions
V(1) = 0;              % Start deflated
P(1) = 0;              % Atmospheric pressure (gauge)
Q(1) = Q_nominal;      % Assume flow rate at t=0: pump nominal flow rate

% Simulation loop: simple Euler method
for i = 2:length(t)
    % Compute back pressure using simplified nonlinear airbag model
    P_back = (V(i-1) >= V_slack) * ... % 1 if V > V_slack
              k * (exp((V(i-1) - V_slack)/V_scale) - 1); % pressure law

    P(i) = P_back;

    % Update flow rate based on back pressure:

```

```

Q(i) = Q_nominal * (P_max - P_back) / P_max;

% Update volume (integrate flow rate):
dV = Q(i) * dt;
V(i) = V(i-1) + dV;

end

%% Find key time points
% Time to reach threshold

idx_thresh = find(V >= V_thresh,1);
t_thresh = t(idx_thresh);
fprintf('Time to reach threshold volume (%.2f L): %.2f sec\n',...
    V_thresh, t_thresh);

% Compute V_predetermined(t_predetermined):
idx_predetermined = find(t >= t_predetermined,1);
V_predetermined = V(idx_predetermined);
fprintf('Predetermined time: %.2f sec\n', t_predetermined);
fprintf('Volume at predetermined time: %.2f L\n', V_predetermined);
fprintf('Asymptotic volume (model limit): %.2f L\n',...
    V_asymptote);

%% Plotting
clf

plot(t(1:idx_predetermined), V(1:idx_predetermined), 'b-', 'LineWidth', 3);
hold on
plot(t(idx_predetermined+1:end), V(idx_predetermined+1:end), 'b--', 'LineWidth', 1.5);

darkgreen = [0, 0.6, 0];
% Add reference lines
yline(V_thresh, 'Color', darkgreen, 'LineStyle','--',...
    'LineWidth', 1.5, 'Label', 'Gas Volume Threshold',...
    'LabelHorizontalAlignment','left');
yline(V_asymptote, 'k:', 'LineWidth', 2.0, ...
    'Label', 'Asymptotic Volume (model limit)',...
    'LabelHorizontalAlignment','left');
yline(V_predetermined, 'm--', 'LineWidth', 1.5,...
    'Label', 'Predetermined Volume', 'LabelHorizontalAlignment','left');
xline(t_predetermined, 'm--', 'LineWidth', 1.5, ...
    'Label', 'Predetermined Time (stopping time)', 'LabelVerticalAlignment','bottom');

% Add reference points
text(t_thresh, V_thresh-0.05, sprintf('t = %.2f sec', t_thresh), ...
    'HorizontalAlignment', 'left',...
    'FontSize', 10, 'BackgroundColor', 'white');
plot(t_thresh, V_thresh, 'o', 'Color', darkgreen, ...
    'MarkerSize', 10, 'MarkerFaceColor', darkgreen);
text(t_predetermined, V_predetermined-0.05, ...
    sprintf('V = %.2f L', V_predetermined),...
    'HorizontalAlignment', 'left',...
    'FontSize', 10, 'BackgroundColor', 'white');
plot(t_predetermined, V_predetermined, ...
    'mo', 'MarkerSize', 10, 'MarkerFaceColor', 'm');

% Show initial slope (dV/dt)
t_linear = t(1:min(10, length(t)));
V_linear = Q_nominal * t_linear;

```

```
plot(t_linear, V_linear, 'k--', 'LineWidth', 1);
text(10, 0.5, sprintf('Initial Flow Rate = %.2f L/min', Q_nominal*60), ...
     'FontSize', 10, 'BackgroundColor', 'white');

xlabel('Time [sec]');
ylabel('Airbag Volume [L]');
title('Simplified Simulation of Airbag Inflation: Volume vs Time');
%axis([0 t_max 0 2.1])
grid on;

orient landscape
print(gcf, '-dpdf', '-fillpage', 'airbag.pdf')
```

Fig. 1 Simplified Simulation of Airbag Inflation: Volume vs Time

