

```

1 % Author: Noah J. Cowan
2 % Date: Oct 29, 2025
3 %% Updated header =====
4 % Revision Date: Aug, 10, 2026
5 %% =====
6 % Airbag Inflation Simulation
7 % Models pump filling airbag with back-pressure effects
8 % This script is illustrative, and not meant to be a detailed mechanistic
9 % simulation. The simplifying assumptions are listed below.
10 clear all; close all;
11 %% Parameters
12 % This parameter set represents a simplified buoyancy control system
13 % for autonomous pool cleaning robots, based on practical considerations.
14 % Pump characteristics (illustrative):
15 Q_nominal = 4.0/60;
16 % Assumed nominal pump flow rate [L/sec]
17 P_max = 50;
18 % Maximum pressure of pump [kPa]
19 % Airbag characteristics (illustrative):
20 V_slack = 1;% Slack (rest) volume [L] - membrane just taut above this
21 V_scale = 1;% Characteristic stretch volume [L] (sets the "knee")
22 k = 10.0;% Nonlinear coefficient [kPa]
23
24 % Design parameters (illustrative)
25 V_thresh = 1.75;% Minimum gas volume threshold for system buoyancy
26 % Assumption: V > V_thresh ==> the robot floats.
27
28 t_predetermined = 40;
29 % Assume a predetermined time of 40 seconds. Time
30 % selected to achieve a predetermined final volume
31 % that exceeds the volume threshold i.e.
32 % V_predetermined > V_thresh
33
34 % Straightforward calculation of the asymptotic volume
35 V_asymptote = V_slack + V_scale * log(1 + P_max/k);
36
37 % Simulation parameters for simple Euler integration
38 dt = 0.01;% Time step [sec]
39 t_max = 80;% Maximum simulation time [sec]
40 t = 0:dt:t_max;% Time vector
41 %% Added code: Specify simulation cases =====
42 simulationCases.names={"Cowan's Simplified Model","Starting Volume: 0.1 L", "Starting Volume:
43 0.2 L"...
44 "Pump Performance: 95% of Nominal Q","Pump Performance: 90% of Nominal
45 Q"}; % Names of each simulation case
46
47 simulationCases.colors={[0 0 0],[0.85 0.35 0.35],[1.00 0.60 0.60],[0.35 0.35 0.85],[0.60 0.60
48 1.00]}; % Plot colors for each simulation case
49
50 simulationCases.startingVolumes=[0, 0.1, 0.2, 0, 0];% Starting volumes [L] for each
51 simulation case
52
53 simulationCases.pumpPerformance=[1, 1, 1, 0.95, 0.9];% Percentage of the nominal flow rate of
54 the pump
55
56 %% =====
57 %% Added Code: Sweep through multiple simulation cases=====
58
59 for simIndex=1:numel(simulationCases.names)

```

```

57
58 %% Initialize arrays
59 V = zeros(size(t));% Volume [L]
60 P = zeros(size(t));% Pressure [kPa]
61 Q = zeros(size(t));% Actual flow rate [L/sec]
62
63 %% Added code:Consider pump performance (reduction in Q_nominal)=====
64
65 %Set pump performance based on simulation case
66 generalPumpPerformance=simulationCases.pumpPerformance(simIndex);
67
68 % Adjust nominal flow rate based on pump performance
69 Q_adjusted = Q_nominal * generalPumpPerformance;
70
71
72 %% =====
73
74
75 % Initial conditions
76 %% Edited Code: Initialize bag as specified starting volume =====
77 %V(1) = 0;% Start deflated
78 V(1)=simulationCases.startingVolumes(simIndex); %Starting volume specified in the simulation
    case
79 %% =====
80 P(1) = 0;% Atmospheric pressure (gauge)
81 Q(1) = Q_nominal;% Assume flow rate at t=0: pump nominal flow rate
82
83
84 %% Simulation loop: simple Euler method
85 for i = 2:length(t)
86 % Compute back pressure using simplified nonlinear airbag model
87 P_back = (V(i-1) >= V_slack) * ...% 1 if V > V_slack
88 k * (exp((V(i-1) - V_slack)/V_scale) - 1); % pressure law
89 P(i) = P_back;
90
91
92
93 %% Added code:Consider pump performance =====
94
95 % Previous method to update flow rate based on back pressure:
96 %Q(i) = Q_nominal * (P_max - P_back) / P_max;
97 % Updated method to consider pump performance
98 Q(i) = Q_adjusted * (P_max - P_back) / P_max;
99
100 %% =====
101 % Update volume (integrate flow rate):
102 dV = Q(i) * dt;
103 V(i) = V(i-1) + dV;
104
105 end
106 %% Find key time points
107 % Time to reach threshold
108 idx_thresh = find(V >= V_thresh,1);
109 t_thresh = t(idx_thresh);
110 fprintf('Time to reach threshold volume (%.2f L): %.2f sec\n',...
111 V_thresh, t_thresh);
112 % Compute V_predetermined(t_predetermined):
113 idx_predetermined = find(t >= t_predetermined,1);
114 V_predetermined = V(idx_predetermined);
115 fprintf('Predetermined time: %.2f sec\n', t_predetermined);
116 fprintf('Volume at predetermined time: %.2f L\n', V_predetermined);

```

```

117 fprintf('Asymptotic volume (model limit): %.2f L\n',...
118 V_asymptote);
119 %% Plotting
120 %% Edited code: commented out clf command so a single figure could display multiple plots
121 % clf
122 %% =====
123
124 %% Edited Code: Plot with colors/label for each simulation case=====
125 %Previous plotting code
126 % plot(t(1:idx_predetermined), V(1:idx_predetermined), 'b-', 'LineWidth', 3);
127 % hold on
128 % plot(t(idx_predetermined+1:end), V(idx_predetermined+1:end), 'b--', 'LineWidth', 1.5);
129
130 %Updated plotting code
131 color=cell2mat(simulationCases.colors(simIndex));
132 name=cell2mat(simulationCases.names(simIndex));
133
134 plot(t(1:idx_predetermined), V(1:idx_predetermined),'Color',color,'LineStyle','-','LineWidth',
135 3, 'DisplayName',name);
136 hold on
137 plot(t(idx_predetermined+1:end), V(idx_predetermined+1:end),'Color', color,'LineStyle','--',
138 'LineWidth', 1.5, 'HandleVisibility','off');
139 % Add legend
140 legend('Location','northwest')
141 %% =====
142
143 %% Edited code: Only plot reference lines on last simulation case=====
144
145 if simIndex==numel(simulationCases.startingVolumes)
146 darkgreen = [0, 0.6, 0];
147 % Modified to set HandleVisibility to off (to omit lines from legend)
148 yline(V_thresh, 'Color', darkgreen, 'LineStyle','--',...
149 'LineWidth', 1.5, 'Label', 'Gas Volume Threshold',...
150 'LabelHorizontalAlignment','left','HandleVisibility','off');
151
152 xline(t_predetermined, 'm--', 'LineWidth', 1.5, ...
153 'Label', 'Predetermined Time (stopping time)',
154 'LabelVerticalAlignment','bottom','HandleVisibility','off');
155 end
156 %% =====
157
158 %% Edited Code: Removed Asymptotic and predetermined volume line =====
159
160 % Add reference lines
161 %yline(V_asymptote, 'k:', 'LineWidth', 2.0, ...
162 %'Label', 'Asymptotic Volume (model limit)',...
163 %'LabelHorizontalAlignment','left');
164 %% =====
165
166 %% Edited Code: Updated predetermined volume line plotter =====
167
168 %Original predetermined volume line plot
169 %yline(V_predetermined, 'm--', 'LineWidth', 1.5,...
170 '%Label', 'Predetermined Volume', 'LabelHorizontalAlignment','left');
171
172 %Updated predetermined volume line plot
173 if simIndex==1
174 yline(V_predetermined,'m--', 'LineWidth', 1.5,'HandleVisibility','off',...
175 'Label', 'Predetermined Volume', 'LabelHorizontalAlignment','left');
176 end

```

```

175 %% =====
176
177 %% Edited code: Removed point labels and text (for clutter) =====
178 % Add reference points
179 % text(t_thresh, V_thresh-0.05, sprintf('t = %.2f sec', t_thresh), ...
180 % 'HorizontalAlignment', 'left',...
181 % 'FontSize', 10,'BackgroundColor', 'white');
182 % plot(t_thresh, V_thresh, 'o', 'Color', darkgreen, ...
183 % 'MarkerSize', 10, 'MarkerFaceColor', darkgreen);
184 % text(t_predetermined, V_predetermined-0.05, ...
185 % sprintf('V = %.2f L', V_predetermined),...
186 % 'HorizontalAlignment', 'left',...
187 % 'FontSize', 10,'BackgroundColor', 'white');
188 % plot(t_predetermined, V_predetermined, ...
189 % 'mo', 'MarkerSize', 10, 'MarkerFaceColor', 'm');
190 % % Show initial slope (dV/dt)
191 % t_linear = t(1:min(10, length(t)));
192 % V_linear = Q_nominal * t_linear;
193 % plot(t_linear, V_linear, 'k--', 'LineWidth', 1);
194 % text(10, 0.5, sprintf('Initial Flow Rate = %.2f L/min', Q_nominal*60), ...
195 % 'FontSize', 10, 'BackgroundColor', 'white');
196 %% =====
197
198
199 xlabel('Time [sec]');
200 ylabel('Airbag Volume [L]');
201 title('Simplified Simulation of Airbag Inflation: Volume vs Time');
202 %axis([0 t_max 0 2.1])
203 grid on;
204 orient landscape
205 %% Edited code: close the simulation case loop =====
206 end
207 %% =====
208 print(gcf, '-dpdf', '-fillpage', 'airbag.pdf')

```

