Problem 2.6 (4pts): Dynamics with Matlab and Simulink

(a) From the transfer function,
\[ H(s) = \frac{Y(s)}{X(s)} = \frac{1}{s^2 + 2s + 4} \]
we have,
\[ s^2 Y'(s) + 2s Y'(s) + 4Y'(s) = X(s) \]

Taking the inverse Laplace transform, the time domain expression becomes:
\[ \dot{y}_1(t) + 2\dot{y}_1(t) + 4y_1(t) = x(t), \]
with initial values \( y_1(0) = \dot{y}_1(0) = 0 \).

Defining the two state variables to be \( y_1(t) \) and \( y_2(t) = \dot{y}_1(t) \), we can write
\[
\begin{cases}
\dot{y}_1(t) = y_2 \\
\dot{y}_2(t) = -4y_1(t) - 2y_2(t) + x(t)
\end{cases}
\]

The system is subjected to a step input \( x(t) = u(t) \), which can be expressed as:
\[
u(t) = \begin{cases} 
0 & x \leq 0 \\
1 & x > 0
\end{cases}
\]

We can create an ode function called \( \text{fnc} \) as follows:

```matlab
function dy = fnc(t,y)
dy = zeros (2,1) ; 
% create a column vector of all zeros
% to define the [y1; y2] array
dy(1) = y(2) ; 
% then define the state derivatives
dy(2) = -4*y(1) - 2*y(2) + 1;
```

Matlab command \( \text{ode45} \) can be used to integrate these equations for a time interval of 10 seconds:

\[
[T,Y] = \text{ode45}(@fnc,[0 10],[0 0]);
\]

Figure 1 shows the system response of \( y_1(t) \), which corresponds to an underdamped 2\textsuperscript{nd}-order system (overshoot, oscillation, etc.). Alternatively, we can integrate using \( \text{ode23} \). The difference between \( \text{ode23} \) and \( \text{ode45} \) is the algorithm: \( \text{ode23} \) uses a lower order algorithm and is more efficient for a coarse estimate. In this simple case, there is not much difference using either algorithm. But whenever you use numerical methods, it is wise to choose the integration parameters (tolerance, for example) to ensure an accurate representation of the system and a practically acceptable computation time.

The full MATLAB script for the problem is posted at the end of the solution. Note that the program \( \text{fnc.m} \) (which defines the function \( \text{fnc} \)) and the program \( \text{p2_6a.m} \) must be in the same directory when \( \text{p2_6a.m} \) is called.

Cite as: Carol Livermore and Joel Voldman, course materials for 6.777J Design and Fabrication of Microelectromechanical Devices, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu/), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
(b) Using tf object

\[ H = \text{tf}(1,[1 \ 2 \ 4]) \]

we create the transfer function corresponding to \( \frac{1}{s^2 + 2s + 4} \), by including the coefficients of the powers of \( s \), as arrays in descending order (of the powers), for the numerator and denominator respectively.

Then we use the MATLAB commands

\[
\text{step}(H) \\
\text{bode}(H)
\]

to plot the step response of the above transfer function and its Bode plot. The step response is the same as Figure 1; the bode plot is shown in Figure 2.

(c) Using SIMULINK

The system can also be represented using a Simulink block diagram as shown in Figure 3 (here using the transfer function formulation). The resulting plot is again similar to that in Figure 1.
Problem 3.8 (2 pts): KOH etched diaphragm

Figure 4 shows the desired diaphragm features on a (100) silicon wafer. The mask feature size $a$ can be expressed as a function of wafer thickness $t_w$, diaphragm thickness $t_d$, diaphragm dimension $d$ and the intersection angle $\theta$ between {111} and {100} planes:

$$ a = d + 2(t_w - t_d)\cot \theta $$

(1)

Substituting the numbers, we get $a = 1079.7 \mu m$.

If the pattern is misaligned by $\theta = 1^\circ$, the actual size of the KOH pit will be $a(\cos \theta + \sin \theta)$ and hence the edge length variation will be

$$ \Delta a = a(\cos \theta + \sin \theta - 1) = 18.7 \mu m $$

(2)

This in turn translates to the edge-length variation for the diaphragm $\Delta d$.

![Figure 4. Cross section of a (100) silicon wafer placed in an anisotropic etchant.](image)

If the sensitivity $S$ of a pressure sensor varies as the inverse fourth power of the diaphragm edge length $d$, i.e.

$$ S \propto d^{-4} $$

then the percentage variation attributed to variations in wafer thickness is:

$$ \% \frac{\Delta S}{S} \propto 4d^{-1} \cdot \Delta d \cdot 100\% $$

(3)

$$ = 18.7\% $$
Problem 4.15 (2 pts): Crayon engineering: Debug and recreate a process and mask set for a pressure-sensing silicon diaphragm

<table>
<thead>
<tr>
<th>Proposed process step</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Start with a double-side-polished n-type silicon wafer.</td>
<td>Must precede with a clean (RCA or piranha)</td>
</tr>
<tr>
<td>2. Perform photolithography using 1-μm-thick positive photoresist to define the diaphragm area</td>
<td>1 μm of PR is very thin when used as a mask in DRIE. In other words, since the selectivity of DRIE to silicon over PR is ~ 50:1, etching through ~ 500 μm of Si would require more than 10 μm of PR. Also, DRIE would lead to a non-uniform membrane thickness with variations on the order of the required thickness (15 μm). This will make the device function improperly if fabricated at all.</td>
</tr>
<tr>
<td>3. Deep-reactive-ion etch the silicon to form the diaphragm; ash resist.</td>
<td></td>
</tr>
<tr>
<td>4. Anodically bond the silicon wafer to a pyrex wafer</td>
<td>Must precede with wafer cleaning</td>
</tr>
</tbody>
</table>

Corrected process:

1. Start with double-side polished SOI wafer, device layer 15 μm thick, oxide layer 1 μm thick, substrate 500 μm thick. RCA clean.
2. Deposit LPCVD nitride, 0.5 μm thick (will coat both sides).
3. Spin 1-μm-thick positive photoresist on bottom side and perform photolithography using Mask 1 to the bottom side.
4. Dry etch the nitride on the bottom side using CF₄/H₂ plasma for example. Ash resist.
5. KOH etch the silicon from the bottom side using the embedded oxide layer as an etch stop. If the dimensions of Mask 1 were calculated correctly, the resultant profile on the top side must be 1 mm across.
6. Piranha clean to remove all resist residue.
7. Etch the remaining nitride in 85% phosphoric acid.
8. Etch the exposed oxide using BOE for ~10 minutes. RCA clean.
9. Anodically bond the patterned SOI wafer to a Pyrex wafer.

The device cross sections at different steps and the mask needed are shown in Figure 5 below. The side of the square on the mask is calculated from equation 1 in Problem 3.8 above.
Figure 5: Corrected process flow and mask for a pressure-sensing silicon diaphragm
Problem 4.13 (4 pts): Crayon engineering: Create process and mask set for a DEP trap

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Starting Material: Pyrex wafer</strong></td>
<td>4” or 6”; will be used as insulating layer B</td>
</tr>
<tr>
<td>1</td>
<td><strong>Clean</strong></td>
</tr>
<tr>
<td>2</td>
<td><strong>Photolithography</strong></td>
</tr>
<tr>
<td>3</td>
<td><strong>Deposit Au-Ti bilayer</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>Lift-off Au-Ti bilayer</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>Clean</strong></td>
</tr>
<tr>
<td>6</td>
<td><strong>Deposit silicon oxide</strong></td>
</tr>
<tr>
<td>7</td>
<td><strong>Photolithography</strong></td>
</tr>
<tr>
<td>8</td>
<td><strong>Deposit Au-Ti bilayer</strong></td>
</tr>
<tr>
<td>9</td>
<td><strong>Lift-off Au-Ti bilayer</strong></td>
</tr>
<tr>
<td>10</td>
<td><strong>Photolithography</strong></td>
</tr>
<tr>
<td>11</td>
<td><strong>Etch oxide</strong></td>
</tr>
<tr>
<td>12</td>
<td><strong>Strip resist</strong></td>
</tr>
<tr>
<td>13</td>
<td><strong>Clean</strong></td>
</tr>
</tbody>
</table>
Figure 6: Process flow and mask set for a DEP trap

Cite as: Carol Livermore and Joel Voldman, course materials for 6.777J Design and Fabrication of Microelectromechanical Devices, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu/), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].
MATLAB CODE FOR PROBLEM 2.6

```matlab
%-----------------------------fnc.m-----------------------------------------
%Create a function fnc to define the relations between the state
%derivatives and the state variables. fnc is then called using ode45.

function dy = fnc(t,y)
    dy = zeros (2,1)        % create a column vector of all zeros
    % to define the [y1;y2] array
    dy(1) = y(2) ;        % then define the state derivatives
    dy(2) = -4*y(1) - 2*y(2) + 1;

%----------------------------END OF PROGRAM----------------------------------

%-----------------------------p2_6a.m----------------------------------------
%MATLAB code for Problem 2.6a
%Use the code below to call the defined function fnc and plot the response
%over 10 sec

[T,Y] = ode45(@fnc,[0 10],[0 0]);
plot(T,Y(:,1))
title('Figure 1. Problem 2.6 Step Response Plot');
xlabel('Time t');
ylabel('Response y_1(t)');

%----------------------------END OF PROGRAM----------------------------------

%-----------------------------p2_6b.m----------------------------------------
%MATLAB code for Problem 2.6b

H=tf(1,[1 2 4]);    % Use tf to create the transfer function
step(H)             % Use step to plot the step response
figure             % Open a new figure
bode(H)             % Use bode to generate the bode plot
title('Figure 2. Problem 2.6b Bode Plot');

%----------------------------END OF PROGRAM----------------------------------
```

Cite as: Carol Livermore and Joel Voldman, course materials for 6.777J Design and Fabrication of Microelectromechanical Devices, Spring 2007. MIT OpenCourseWare (http://ocw.mit.edu/), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].