EE236: Experiment 5

PMOS I/V Characteristics and Applications

Param Rathour, 190070049

Spring Semester, 2021-22

Contents

1 Overview of the experiment

This report contains my approach to the experiment, the circuit's design with the relevant simulation code and output plots.

1.1 Aim of the experiment

- To plot and understand I_D/V_{DS} characteristics of PMOS at different V_{GS} and calculate r_{DS} (resistance linear region), r_0 (resistance saturation region) and V_A (early voltage) considering channel length modulation.
- To plot and understand I_D/V_{GS} characteristics of PMOS in different regions and obtain the parameters V_T, g_m and K (in mA/V^2)
- To plot and understand the effect of Body Bias on I_D/V_{GS} characteristics of PMOS and calculate γ (body effect coefficient)

1.2 Methods

First, I_D/V_{DS} characteristics of PMOS at different V_{GS} was plotted and the parameters were calculated & compared.
Then, I_D/V_{GS} characteristics of PMOS at was plotted in different regions by changing V_{DS} and the parameters were calculated.
Lastly, I_D/V_{GS} characteristics of PMOS at was plotted for different body voltage and behaviour of V_T verses V_{SB} was observed which was used to calculate body effect coefficient.
All parameters were calculated using python by exporting the vectors generated by NGSPICE.

2 Design

The common circuit designed for all parts is shown below

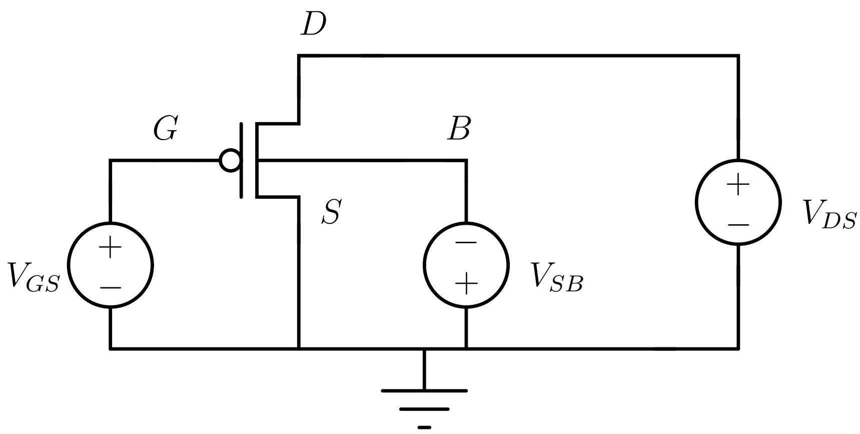


Figure 1: Circuit for I/V measurement

2.1 I_D/V_{DS} Characteristics of PMOS

For the first part the Voltage V_{GS} was varied from $-2.5V$ to $-4V$ in steps of $-0.5V$.
For each V_{GS}, V_{DS} was varied from 0 to $-5V$
As in linear region slope I_D/V_{DS} is maximum $\rightarrow r_{DS} = \frac{1}{\text{maximum slope}}$
Similarly in saturation region slope I_D/V_{DS} is minimum $\rightarrow r_0 = \frac{1}{\text{minimum slope}}$
Early voltage can be calculated by taking the mean of the $x-$intercepts of the extrapolated lines from saturation regions.

$$x - \text{intercept} = \frac{-(y - \text{intercept})}{slope}$$

and slope, $y-$intercepts were calculated using Python's `Scipy` package This graph is a straight line in some range of I, V. Also, $I_L = 0$ for Dark characteristics.
We can calculate its slope to get η and y-intercept by interpolating that straight line.

2.2 I_D/V_{GS} characteristics of PMOS for different V_{DS}

Again, the above circuit was used (Figure 1), V_{GS} was varied from 0 to $-5V$

$$I_D = \frac{1}{2}K(2(V_{GS} - V_T)V_{DS} - (V_{DS})^2)$$

In linear region, $(V_{DS})^2$ can be ignored (as $V_{DS} = -200mV$) to give

$$I_D = K(V_{GS} - V_T)V_{DS}$$

So $V_T = -(\underbrace{y - \text{intercept}}_{-K\cdot V_T\cdot V_{DS}})/\underbrace{\text{slope}}_{K\cdot V_{DS}}$ and $g_m = \frac{\partial I_D}{\partial V_{GS}} = \frac{I_D}{V_{GS}}$(as the plot is linear)
In saturation region, as $V_{DS} = -5V \leq (V_{GS} - V_T)$, I_D depends only on $(V_{GS} - V_T)$ (ideally)

$$I_D = \frac{1}{2}K(V_{GS} - V_T)^2$$

$$\sqrt{I_D} = \frac{1}{\sqrt{2}}\sqrt{K}(V_{GS} - V_T)$$

So from the $\sqrt{I_D}/V_{GS}$ characteristics, $K = 2 \cdot \text{slope}^2$ (in A/V^2) $= 1000 \cdot 2 \cdot \text{slope}^2$ (in mA/V^2)
$V_T = -(\underbrace{y - \text{intercept}}_{-\frac{1}{\sqrt{2}}\sqrt{K}\cdot V_T})/\underbrace{\text{slope}}_{\frac{1}{\sqrt{2}}\sqrt{K}}$
and $g_m = \frac{\partial I_D}{\partial V_{GS}}$ will be different for different V_{GS} (as the plot is quadratic)
Say, $V_{GS} = -3V$, then at that point, $g_m = \frac{I_D}{V_{GS}}\mho$

2.3 I_D/V_{GS} characteristics of PMOS for different V_{SB}

Again, the same circuit ((Figure 1)) was used.
With V_{SB} varying from 0 to $-4V$ in steps of $-1V$. $V_{DS} = -200mV$
To calculate V_T, again we extrapolate the lines and get the $x -$ intercept.
Then γ was calculated for each V_T and V_{SB} as shown below (average of all γ was taken)

$$V_T = V_{T0} + \gamma(\sqrt{\varphi_s - V_{SB}} - \sqrt{\varphi_s}) \quad (\text{where } \varphi_s = 0.8V)$$

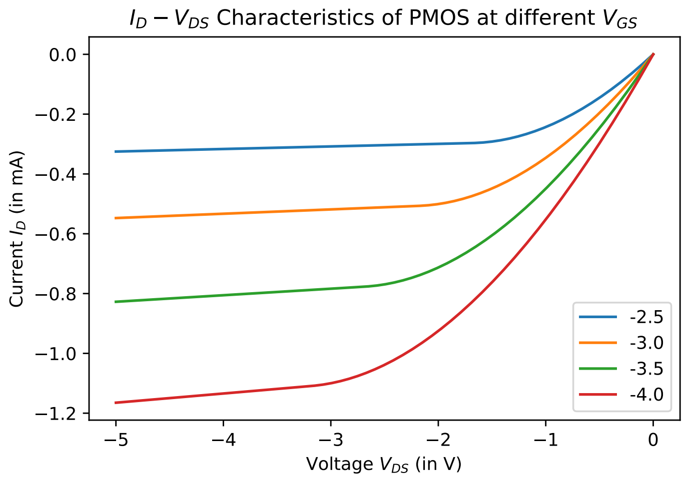
3 Simulation results

3.1 Plots

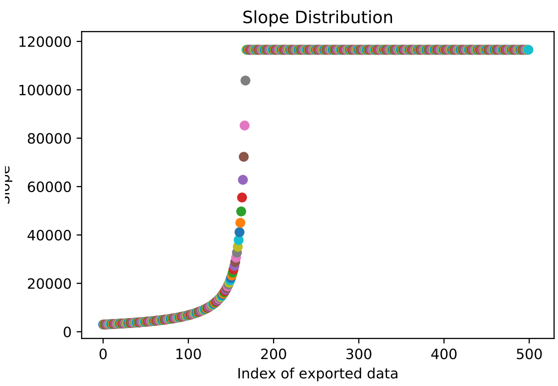
3.1.1 I_D/V_{DS} characteristics of PMOS for different V_{GS}

V_{GS} (in V)	R_{DS} (in Ω)	V_A (in V)	R_0 (in Ω)
-2.5	2984.17	32.89471976	116548.76
-3	2298.15	32.8947267	69217.08
-3.5	1868.59	32.89473013	45799.08
-4	1574.32	32.89436244	32528.79

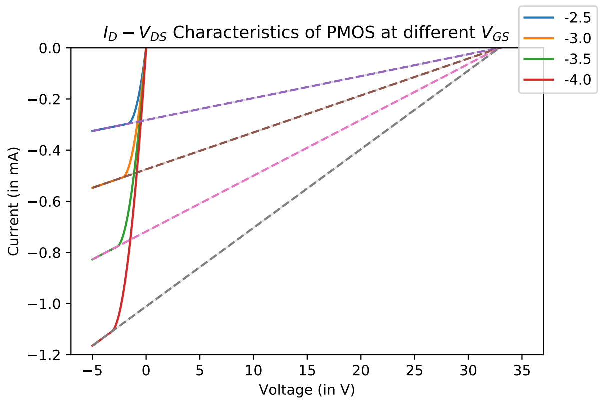
Table 1: Parameter Values obtained from the characteristics



(a) Different V_{GS}



(b) Slope Distribution

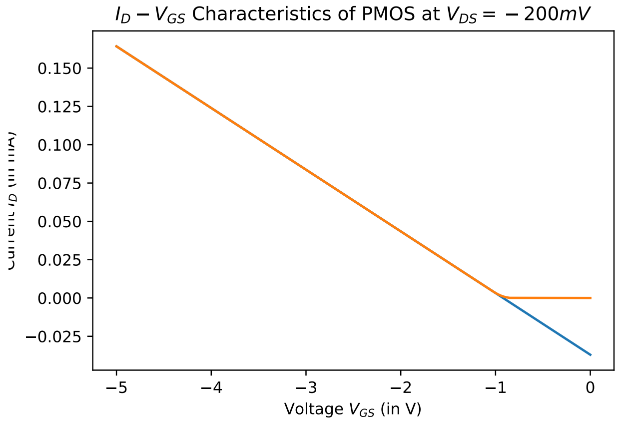


(c) Characteristics with the Extrapolated Lines

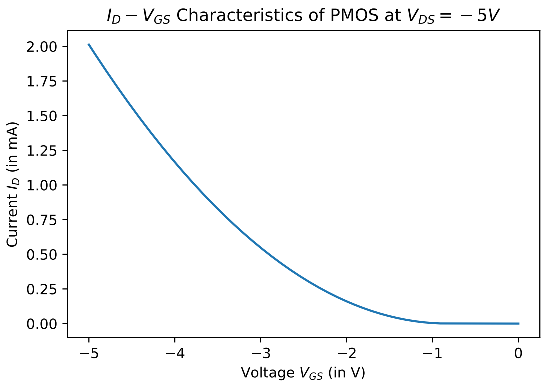
Figure 2: I_D/V_{DS} characteristics of PMOS

In Figure 2, as magnitude of V_{GS} increases, magnitude of I_D increases.
All extrapolated lines meet at $V_A = 32.89463475487444V$

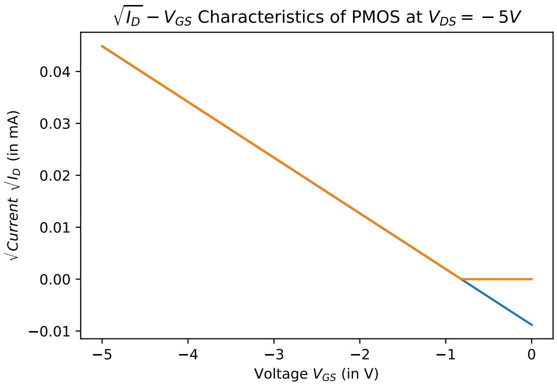
3.1.2 I_D/V_{GS} characteristics of PMOS for different V_{DS}



(a) $V_{DS} = -200mV$ with the Extrapolated Line



(b) $V_{DS} = -5mV$

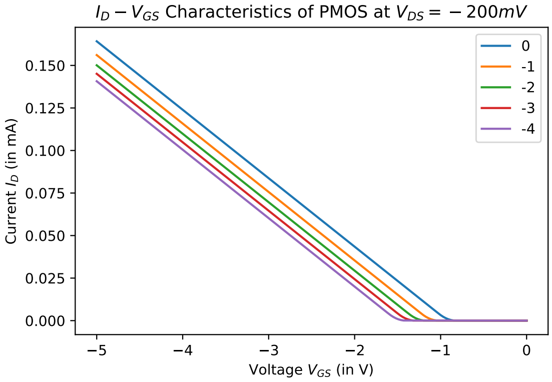


(c) $\sqrt{I_D}/V_{GS}$ characteristics when $V_{DS} = -5V$ with the Extrapolated Line

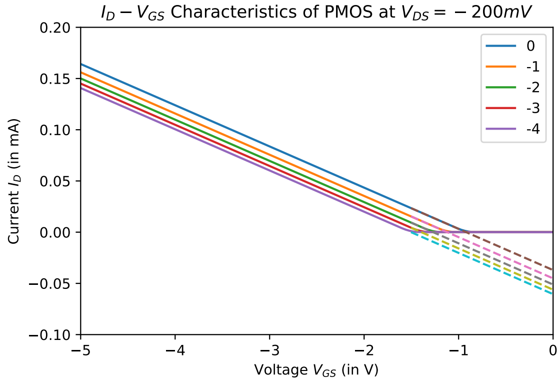
Figure 3: I_D/V_{GS} characteristics of PMOS

In Figure 3a), as magnitude of V_{GS} increases (neglecting the small V_{GS} values as PMOS is saturated), magnitude of I_D increases linearly giving
$V_T = -0.92V$ and $g_m = -0.0402432m\mho$
In Figure 3b), as magnitude of V_{GS} increases (neglecting the small V_{GS} values as PMOS is not saturated), magnitude of I_D increases quadratically giving
$V_T = -0.82V$, $g_m = -0.503424m\mho$ and $K = 0.2304mA/V^2$

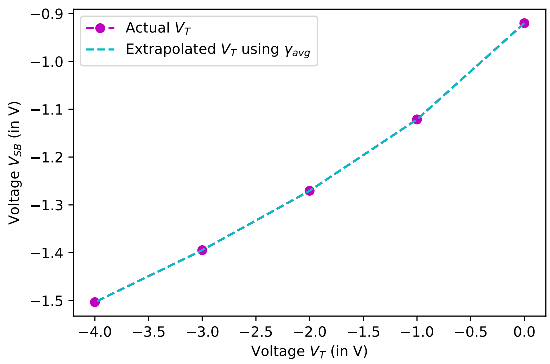
3.1.3 I_D/V_{GS} characteristics of PMOS for different V_{SB}



(a) Different V_{SB}



(b) Extrapolated Lines



(c) V_T vs V_{SB}

Figure 4: I_D/V_{GS} characteristics of PMOS

In Figure 4, as magnitude of V_{SB} increases, magnitude of I_D reduces giving $V_T = -0.92V, -1.1212461V, -1.27050172V, -1.39471918V, -1.50340825V$ for $V_{SB} = 0V, -1V, -2V, -3V, -4V$ respectively.
From these values, we get $\gamma = -0.45\sqrt{V}$ and it fits the original curve very well.
The dependence of V_T on V_{SB} is similar to a square root graph which increases very slowly. As magnitude of V_T increases, V_{SB}'s magnitude decreases.

3.2 NGSPICE Netlists

3.2.1 I_D/V_{DS} characteristics of PMOS for different V_{GS}

3.2.1.1 $V_{GS} = -2.5V$

```
Param Rathour (190070049), I_D - V_DS Characteristics of PMOS at V_GS = -2.5V
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd 0
Vdummy d mid 0
VG g gnd -2.5
.dc VD 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = -I(Vdummy)
let V_DS = V(d)
plot I_D vs V_DS
wrdata 111.txt I_D vs V_DS
.endc
.end
```

3.2.1.2 $V_{GS} = -3V$

```
Param Rathour (190070049), I_D - V_DS Characteristics of PMOS at V_GS = -3V
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd 0
Vdummy d mid 0
VG g gnd -3
.dc VD 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = -I(Vdummy)
let V_DS = V(d)
plot I_D vs V_DS
wrdata 112.txt I_D vs V_DS
.endc
.end
```

3.2.1.3 $V_{GS} = -3.5V$

```
Param Rathour (190070049), I_D - V_DS Characteristics of PMOS at V_GS = -3.5V
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd 0
```

```
Vdummy d mid 0
VG g gnd -3.5
.dc VD 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = -I(Vdummy)
let V_DS = V(d)
plot I_D vs V_DS
wrdata 113.txt I_D vs V_DS
.endc
.end
```

3.2.1.4 $V_{GS} = -4V$

```
Param Rathour (190070049), I_D - V_DS Characteristics of PMOS at V_GS = -4V
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd 0
Vdummy d mid 0
VG g gnd -4
.dc VD 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = -I(Vdummy)
let V_DS = V(d)
plot I_D vs V_DS
wrdata 114.txt I_D vs V_DS
.endc
.end
```

3.2.2 I_D/V_{GS} characteristics of PMOS for different V_{DS}

3.2.2.1 $V_{DS} = -200mV$

```
Param Rathour (190070049), I_D - V_GS Characteristics of PMOS at V_DS = -200mV
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
```

```
wrdata 211.txt I_D vs V_GS
.endc
.end
```

3.2.2.2 $V_{DS} = -5V$

```
Param Rathour (190070049), I_D - V_GS Characteristics of PMOS at V_DS = -5V
.include pmos.txt ; Includes PMOS Model
M1 d g gnd gnd ALD1107
VD mid gnd -5
Vdummy d mid 0
VG g gnd 0
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot sqrt(I_D) vs V_GS
wrdata 221.txt I_D vs V_GS
.endc
.end
```

3.2.3 I_D/V_{GS} characteristics of PMOS for different V_{SB}

3.2.3.1 $V_{SB} = 0$

```
Param Rathour (190070049), Effect of Body Bias
.include pmos.txt ; Includes PMOS Model
M1 d g gnd b ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
VB b gnd 0
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
wrdata 312.txt I_D vs V_GS
.endc
.end
```

3.2.3.2 $V_{SB} = -1V$

```
Param Rathour (190070049), Effect of Body Bias
.include pmos.txt ; Includes PMOS Model
M1 d g gnd b ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
VB b gnd 1
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
wrdata 312.txt I_D vs V_GS
.endc
.end
```

3.2.3.3 $V_{SB} = -2V$

```
Param Rathour (190070049), Effect of Body Bias
.include pmos.txt ; Includes PMOS Model
M1 d g gnd b ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
VB b gnd 2
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
wrdata 313.txt I_D vs V_GS
.endc
.end
```

3.2.3.4 $V_{SB} = -3V$

```
Param Rathour (190070049), Effect of Body Bias
.include pmos.txt ; Includes PMOS Model
M1 d g gnd b ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
```

```
VB b gnd 3
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
wrdata 314.txt I_D vs V_GS
.endc
.end
```

3.2.3.5 $V_{SB} = -4V$

```
Param Rathour (190070049), Effect of Body Bias
.include pmos.txt ; Includes PMOS Model
M1 d g gnd b ALD1107
VD mid gnd -200m
Vdummy d mid 0
VG g gnd 0
VB b gnd 4
.dc VG 0 -5 -0.01           ; DC Analysis
.control
run
let I_D = I(Vdummy)
let V_GS = V(g)
plot I_D vs V_GS
wrdata 315.txt I_D vs V_GS
.endc
.end
```

3.3 Python Code for Plots and Calculations

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import math
import pandas as pd
import bisect
```

3.3.1 I_D/V_{DS} characteristics of PMOS for different V_{GS}

```python
V_GS = np.arange(-2.5,-4.5,-0.5)

V_DS = []
I_D = []
```

```python
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{DS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{DS}$ Characteristics of PMOS at different $V_{GS}$')

for i in range(len(V_GS)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab5\\11' +  str(i+1)
    + '.txt', header = None, skipinitialspace=True, delim_whitespace=True)
    V_DS.append(data[0])
    I_D.append(data[1])
    ax1.plot(V_DS[i], 1000*I_D[i], '-o', markersize=0.01)
ax1.legend(V_GS)
fig1.set_dpi(150)
fig1.savefig('111.pdf')

slopes = []
for i in range(len(V_GS)):
    slope = []
    for j in range(len(V_DS[i])-1):
        slope.append((I_D[i][j+1]-I_D[i][j])/(V_DS[i][j+1]-V_DS[i][j]))
    slopes.append(slope)
# slopes[0]

slope_min = np.zeros(len(V_GS))
slope_max = np.zeros(len(V_GS))
for i in range(len(V_GS)):
    slope_min[i] = min(slopes[i])
    slope_max[i] = max(slopes[i])
print(slope_min)
print(slope_max)

r_DS = [1/i for i in slope_max]
print("r_DS", r_DS)
r_0= [1/i for i in slope_min]
print("r_0", r_0)

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Index of exported data')
ax1.set_ylabel('Slope')
ax1.set_title('Slope Distribution')
for i in range(len(V_DS[0])-1):
    plt.scatter(i,(V_DS[0][i+1]-V_DS[0][i])/(I_D[0][i+1]-I_D[0][i]))
fig1.set_dpi(150)
fig1.savefig('121.pdf')

y_intercepts = np.zeros(len(V_GS))
```

```
slope_fit = np.zeros(len(V_GS))
for i in range(len(V_GS)):
    slope, intercept, r_value, p_value, std_err =
    stats.linregress(V_DS[i][-2:], I_D[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(V_GS))
R_0 = np.zeros(len(V_GS))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([-1.2,0])
ax1.set_title('$I_D-V_{DS}$ Characteristics of PMOS at different $V_{GS}$')
fig1.set_dpi(150)
intersect = 35
x = np.arange(-5,intersect,0.01)
for i in range(len(V_GS)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o',
    markersize=0.01)
    ax1.plot(V_DS[i], 1000*I_D[i], '-o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
    R_0[i] = 1/slope_fit[i]
print(V_A)
print(np.mean(V_A))
print(R_0)
fig1.savefig('122.pdf')
```

3.3.2 I_D/V_{GS} characteristics of PMOS for different V_{DS}

```
slopes = []
for i in range(len(V_GS)):
    slope = []
    for j in range(len(V_DS[i])-1):
        slope.append((I_D[i][j+1]-I_D[i][j])/(V_DS[i][j+1]-V_DS[i][j]))
    slopes.append(slope)
# slopes[0]

slope_min = np.zeros(len(V_GS))
slope_max = np.zeros(len(V_GS))
for i in range(len(V_GS)):
    slope_min[i] = min(slopes[i])
    slope_max[i] = max(slopes[i])
print(slope_min)
```

```python
print(slope_max)

r_DS = [1/i for i in slope_max]
print("r_DS", r_DS)
r_0= [1/i for i in slope_min]
print("r_0", r_0)

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Index of exported data')
ax1.set_ylabel('Slope')
ax1.set_title('Slope Distribution')
for i in range(len(V_DS[0])-1):
    plt.scatter(i,(V_DS[0][i+1]-V_DS[0][i])/(I_D[0][i+1]-I_D[0][i]))
fig1.set_dpi(150)
fig1.savefig('121.pdf')

y_intercepts = np.zeros(len(V_GS))
slope_fit = np.zeros(len(V_GS))
for i in range(len(V_GS)):
    slope, intercept, r_value, p_value, std_err =
    stats.linregress(V_DS[i][-2:], I_D[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_A = np.zeros(len(V_GS))
R_0 = np.zeros(len(V_GS))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage (in V)')
ax1.set_ylabel('Current (in mA)')
ax1.set_ylim([-1.2,0])
ax1.set_title('$I_D-V_{DS}$ Characteristics of PMOS at different $V_{GS}$')
fig1.set_dpi(150)
intersect = 35
x = np.arange(-5,intersect,0.01)
for i in range(len(V_GS)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o',
    markersize=0.01)
    ax1.plot(V_DS[i], 1000*I_D[i], '-o', markersize=0.01)
    V_A[i] = -y_intercepts[i]/slope_fit[i]
    R_0[i] = 1/slope_fit[i]
print(V_A)
print(np.mean(V_A))
print(R_0)
fig1.savefig('122.pdf')
```

```
data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab5\\211.txt', header =
None, skipinitialspace=True, delim_whitespace=True)
V_GS = data[0]
I_D = data[1]

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of PMOS at $V_{DS}=-200mV$')
# ax1.legend(V_GS)
fig1.set_dpi(150)
slope, y_intercept, r_value, p_value, std_err = stats.linregress(V_GS[-2:],
1000*I_D[-2:])
ax1.plot(V_GS, slope*V_GS + y_intercept, '-o', markersize=0.01)
ax1.plot(V_GS, 1000*I_D, '-o', markersize=0.01)
print(y_intercept)
V_T = -y_intercept/slope
print("V_T =", V_T)
g_m = slope
print("g_m =", g_m)  # in m mhos
fig1.savefig('21.pdf')

data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab5\\221.txt', header =
None, skipinitialspace=True, delim_whitespace=True)
V_GS = data[0]
I_D = data[1]
I_D_sqrt = [math.sqrt(i) for i in I_D]
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of PMOS at $V_{DS}=-5V$')
# ax1.legend(V_GS)
fig1.set_dpi(150)
ax1.plot(V_GS, 1000*I_D, '-o', markersize=0.01)

fig2, ax2 = plt.subplots()
ax2.set_xlabel('Voltage $V_{GS}$ (in V)')
ax2.set_ylabel('$\sqrt{{Current}}$ $\sqrt{I_D}$ (in mA)')
ax2.set_title('$\sqrt{I_D}-V_{GS}$ Characteristics of PMOS at $V_{DS}=-5V$')
slope, y_intercept, r_value, p_value, std_err = stats.linregress(V_GS[-2:],
(I_D_sqrt[-2:]))
ax2.plot(V_GS, slope*V_GS + y_intercept, '-o', markersize=0.01)
ax2.plot(V_GS, I_D_sqrt, '-o', markersize=0.01)
fig2.set_dpi(150)
V_T = -y_intercept/slope
print("V_T =", V_T)
K = 1000*2*slope**2
```

```
print("K =", K) # mA/V^2
# print(V_GS[300])
g_m = 1000*(I_D[301]-I_D[300]) / (V_GS[301]-V_GS[300])
print("g_m =", g_m)  # in m mhos
fig1.savefig('221.pdf')
fig2.savefig('222.pdf')
# fig2.legend(["Characteristics", "Extrapolation"])
```

3.3.3 I_D/V_{GS} characteristics of PMOS for different V_{SB}

```
V_SB = np.arange(0,-5,-1)

V_GS = []
I_D = []

fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_title('$I_D-V_{GS}$ Characteristics of PMOS at $V_{DS}=-200mV$')

for i in range(len(V_SB)):
    data = pd.read_csv('E:\Program_Files\Spice64\EE236\Lab5\\31' +  str(i+1)
    + '.txt', header = None, skipinitialspace=True, delim_whitespace=True)
    V_GS.append(data[0])
    I_D.append(data[1])
    ax1.plot(V_GS[i], 1000*I_D[i], '-o', markersize=0.01)
ax1.legend(V_SB)
fig1.set_dpi(150)
fig1.savefig('311.pdf')

y_intercepts = np.zeros(len(V_SB))
slope_fit = np.zeros(len(V_SB))
for i in range(len(V_SB)):
    slope, intercept, r_value, p_value, std_err =
    stats.linregress(V_GS[i][-2:], I_D[i][-2:])
    y_intercepts[i] = intercept
    slope_fit[i] = slope
print(slope_fit)
print(y_intercepts)

V_T = np.zeros(len(V_SB))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{GS}$ (in V)')
ax1.set_xlim([-5,0])
ax1.set_ylabel('Current $I_D$ (in mA)')
ax1.set_ylim([-0.1,0.2])
```

```
ax1.set_title('$I_D-V_{GS}$ Characteristics of PMOS at $V_{DS}=-200mV$')
fig1.set_dpi(150)
intersect = 10
for i in range(len(V_SB)):
    x = np.arange(-y_intercepts[i]/slope_fit[i],intersect,0.01)
    ax1.plot(V_GS[i], 1000*I_D[i], '-o', markersize=0.01)
    V_T[i] = -y_intercepts[i]/slope_fit[i]
for i in range(len(V_SB)):
    ax1.plot(x, 1000*(slope_fit[i]*x+y_intercepts[i]), '--o',
    markersize=0.01)
print(V_T)
ax1.legend(V_SB)
fig1.savefig('312.pdf')

gamma = []
phi_s = 0.8
V_T0 = V_T[0]
for i in range(1,len(V_SB)):
    gamma.append((V_T[i] - V_T0)/(math.sqrt(phi_s-V_SB[i]) -
    math.sqrt(phi_s)))

gamma_avg = np.mean(gamma)
gamma_avg
V_Ta = np.zeros(len(V_SB))
for i in range(len(V_SB)):
    V_Ta[i] = V_T0 + gamma_avg*(math.sqrt(phi_s-V_SB[i]) - math.sqrt(phi_s))
fig1, ax1 = plt.subplots()
ax1.set_xlabel('Voltage $V_{T}$ (in V)')
ax1.set_ylabel('Voltage $V_{SB}$ (in V)')
ax1.plot(V_SB, V_T, '--om')
ax1.plot(V_SB, V_Ta, '--c')
fig1.set_dpi(150)
plt.legend(["Actual $V_T$", "Extrapolated $V_T$ using $\gamma_{avg}$"])
fig1.savefig('32.pdf')
```

4 Experiment completion status

Completed everything in Lab successfully.

5 Questions for reflection

The dependence of V_T on V_{SB} is similar to a square root graph which increases very slowly. As magnitude of V_T increases, V_{SB}'s magnitude decreases.
Please see my Design and Simulation Result sections for my other comments.
Overall, the lab was good.Thanks!