

PERANCANGAN APLIKASI OP-AMP DENGAN SOFTWARE GUI MATLAB

Triyono

Program Studi Teknik Elektro

Fakultas Teknik

Universitas Muhammadiyah Tangerang

Jl. Perintis Kemerdekaan I/33, Cikokol, Kota Tangerang

E-mail: *lover_mu_1980@yahoo.co.id*

ABSTRAK

Operational Amplifier (OP-AMP) merupakan rangkaian penguat yang menjadi basis dari rangkaian audio dan video amplifier, filter, buffer (penyangga), komparator, oscilator dan rangkaian analog lainnya. Untuk meningkatkan pemahaman mahasiswa dalam merancang rangkaian OP-AMP dapat digunakan media pembelajaran berbantuan komputer. Matlab merupakan software yang digunakan dalam pembelajaran untuk melihat tanggapan beragam parameter dan masukan yang berbeda. Dari hasil penelitian didapatkan hasil yang cukup signifikan dalam peningkatan pemahaman mahasiswa dalam perancangan rangkaian OP-AMP.

Kata Kunci: OP-AMP, Matlab, Analog.

1. PENDAHULUAN

Operational Amplifier (OP-AMP) telah digunakan bertahun-tahun. Awalnya OP-AMP dibentuk menggunakan sirkuit diskrit transistor, setelah perkembangan teknologi *integrated circuit* (IC) telah merevolusi desain rangkaian analog. OP-AMP merupakan yang pertama sebuah IC analog, hal ini dikarenakan fungsinya yang sangat bermanfaat penggunaannya dalam beragam rangkaian.

Popularitas OP-AMP disebabkan oleh fleksibilitasnya. *High-gain* penguat DC yang memiliki input diferensial tegangan output adalah perbedaan tegangan antara dua input dikalikan dengan *gain*. Komponen pasif bisa digunakan untuk memberikan umpan balik, dan ini mengatur gain dan fungsi sirkuit OP-AMP keseluruhan. Komponen umpan balik negatif pasif mengakibatkan linear respon, yaitu *output* sebanding dengan input. Umpan balik positif pasif hasil dalam *switching* atau osilasi. Komponen kadang-kadang aktif seperti transistor dan dioda digunakan dalam

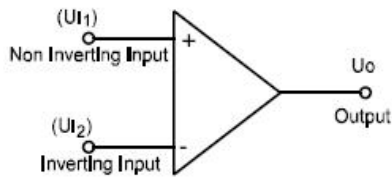
loop umpan balik untuk memberikan respon non-linear.

Permasalahan yang dihadapi dalam perancangan OP-AMP adalah mendapatkan nilai *gain*, *output* dan fungsi grafik rangkaian tersebut. Setelah didapatkan permasalahan selanjutnya adalah menganalisisnya apakah rangkaian yang dibuat berjalan dengan baik atau tidak. Seringkali mahasiswa mengalami kesulitan dalam menganalisis sistem, namun dengan bantuan *software* matlab proses analisis sistem menjadi jauh lebih mudah dan cepat sehingga akan memudahkan dalam proses pembelajaran terutama dalam perancangan sistem dengan menggunakan rangkaian OP-AMP.

2. OPERATIONAL AMPLIFIER

Pada OP-AMP terdapat satu terminal keluaran, dan dua terminal masukan. Terminal masukan yang diberi tanda (-) dinamakan terminal masukan pembalik (*inverting*), sedangkan terminal masukan yang diberi (+)

dinamakan terminal masukan bukan pembalik (*noninverting*). Berikut merupakan gambar simbol OP-AMP.



Gambar 1. Simbol OP-AMP

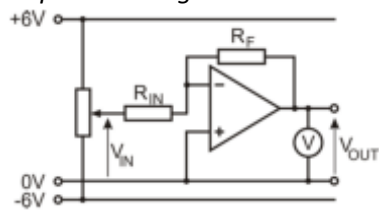
Jika *input* OP-AMP diberi tegangan searah dengan *input Non Inverting* (+) lebih besar dari pada *input inverting* (-), maka pada *output* OP-AMP akan positif (+). Sebaliknya jika *input Non Inverting* (+) lebih kecil dari pada *input inverting* (-), maka *output* OP-AMP akan negatif (-).

Tabel 1. Perbedaan parameter beberapa OP-AMP

Parameter	Ideal	LM741	LF347	LM318
Open-loop Gain (A_{OL})	∞	$2 \cdot 10^5$	10^5	$2 \cdot 10^5$
Input Resistance (R_{in})	$\infty \Omega$	$2 M\Omega$	$10^{12} \Omega$	$3 M\Omega$
Output Resistance (R_o)	0Ω	75Ω	75Ω	75Ω
Gain Bandwidth Product	∞ Hz	1 MHz	4 MHz	15 MHz
CMRR	∞	90 dB	100 dB	100 dB

2.1 Penguat *Inverting*

Rangkaian dasar penguat *inverting* adalah seperti yang ditunjukkan pada gambar 2, dimana sinyal masukannya dibuat melalui *input inverting*.



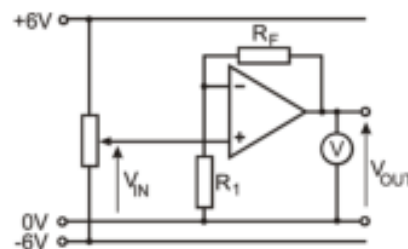
Gambar 2. Penguat *Inverting* Input non-inverting pada rangkaian

ini dihubungkan ke ground, atau $V^+ = 0$. Karena nilainya = 0, namun tidak terhubung langsung ke *ground*, input op-amp V^- pada rangkaian ini dinamakan *virtual ground*. Dengan demikian dapat dihitung tegangan jepit pada R_{in} adalah $V_{in} - V^- = V_{in}$ dan tegangan jepit pada resistor R_f adalah V_{out}

$-v^- = V_{out}$. Kemudian di ketahui bahwa: $i_{in} + i_{out} = i^- = 0$, arus masukan op-amp adalah 0. Maka: $i_{in} + i_{out} = V_{in}/R_{in} + V_{out}/R_f = 0$, Selanjutnya $V_{out}/R_f = -V_{in}/R_{in}$ atau $V_{out}/V_{in} = -R_f/R_{in}$. Jika penguatan G didefinisikan sebagai perbandingan tegangan keluaran terhadap tegangan masukan, maka dapat ditulis $G = V_{out}/R_f = -V_{in}/R_{in}$.

2.2. Penguat *Non-Inverting*

Rangkaian dasar penguat *inverting* adalah seperti yang ditunjukkan pada gambar 3, dimana sinyal masukannya dibuat melalui *input non-inverting*. Dengan demikian tegangan keluaran rangkaian ini akan satu fasa dengan tegangan inputnya.

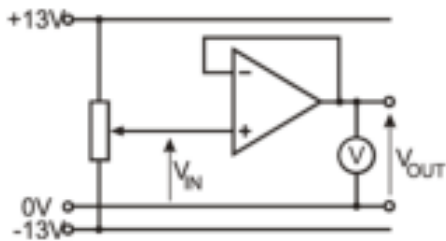


Gambar 3. Penguat *Non-Inverting*

Jika: $V_{in} = V^+$ dan $V^+ = V^- = V_{in}$. Dari sini ketahui tegangan jepit pada R_f adalah $V_{out} - V^- = V_{out} - V_{in}$, atau $i_{out} = (V_{out}-V_{in})/R_f$. Lalu tegangan jepit pada R_1 adalah $V^- = V_{in}$, yang berarti arus $i_{R1} = V_{in}/R_1$. Hukum kirchoff pada titik input inverting bahwa $i_{out} + i(-) = i_{R1}$ serta $i(-) = 0$ dan jika disubsitusi ke rumus yang sebelumnya, maka diperoleh $i_{out} = i_{R1}$ dan Jika ditulis dengan tegangan jepit masing-masing maka diperoleh $(V_{out} - V_{in})/R_f = V_{in}/R_1$ yang kemudian dapat disederhanakan menjadi: $V_{out} = V_{in} (1 + R_f/R_1)$. Jika penguatan G adalah perbandingan tegangan keluaran terhadap tegangan masukan, maka didapat penguatan op-amp *non-inverting*: $G = V_{out} / V_{in} = (1 + R_f/R_1)$.

2.3. Voltage Follower

Rangkaian dasar penguat *voltage follower* adalah seperti yang ditunjukkan pada gambar 4, rangkaian *voltage follower* berguna untuk meningkat arus tanpa mengubah tegangannya. Digunakan untuk mengubah sinyal berimpedansi tinggi menjadi sinyal berimpedansi rendah yang kokoh (robust). Gain *voltage follower* 1.



Gambar 4. Voltage Follower

R_{in} adalah tahanan yang sangat besar dan R_{out} adalah tahanan yang sangat kecil. Kembali ke persamaan awal saat V_{out} sama dengan V_{in} maka persamaan awal menjadi:

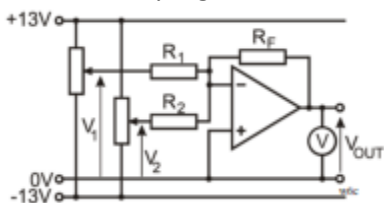
$$V_{out} = A(V^+ - V_{out})$$

sehingga bila V_{out} yang dicari maka persamaan menjadi:

$$V_{out} = \frac{AV^+}{1+A} \approx V^+$$

2.4. Summing Amplifier

Rangkaian dasar penguat *summing amplifier* adalah seperti yang ditunjukkan pada gambar 5, *summing amplifier* digunakan sebagai penjumlah tegangan. *Summing amplifier* dapat digunakan untuk menjumlah tegangan dari beberapa sumber yang memiliki arus yang berbeda.



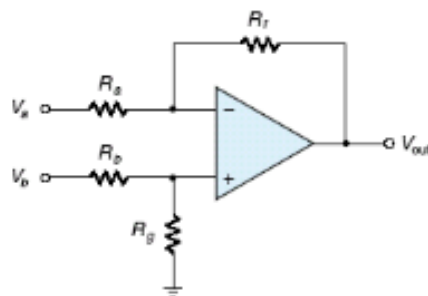
Gambar 5. Summing Amplifier

Persamaan tegangan keluaran ditentukan sebagai berikut:

$$V_{out} = -\left(\frac{R_f}{R_1}V_1 + \frac{R_f}{R_2}V_2\right)$$

2.5. Differential Amplifier

Rangkaian dasar penguat *differential amplifier* adalah seperti yang ditunjukkan pada gambar 6, *differential amplifier* merupakan amplifier yang digunakan untuk mencari selisih tegangan dari dua sinyal yang masuk.



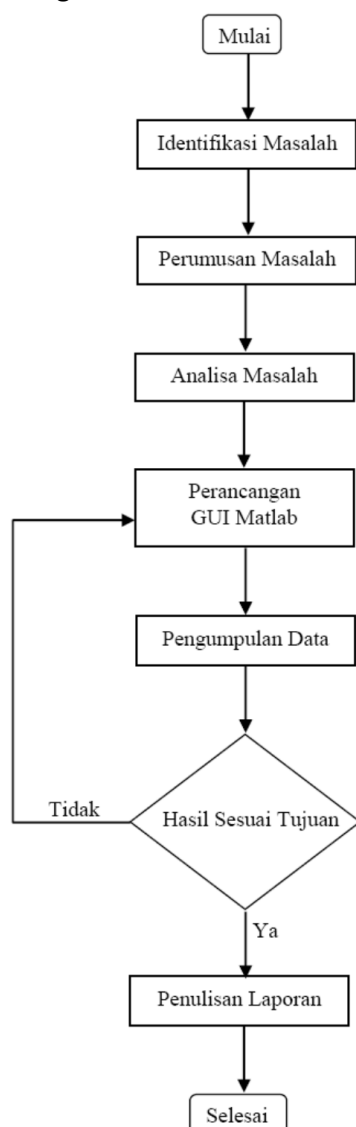
Gambar 5. Differential Amplifier

Persamaan tegangan *output*-nya dinyatakan sebagai berikut:

$$V_{out} = \frac{R_f}{R_a}(V_b - V_a)$$

Tegangan *output* merupakan tegangan tunggal yang mengacu pada *ground* biasanya disebut *single ended voltage*.

3. Metodologi Penelitian



4. Hasil

Rancangan aplikasi GUI Op-Amp diujicobakan kepada mahasiswa dibandingkan dengan menggunakan metode praktikum, metode perhitungan manual. Setelah dilakukan tindakan kelas terjadi peningkatan yang cukup signifikan terhadap pemahaman dan ketertarikan mahasiswa dalam mempelajari materi analog berbasis *operational amplifier* pada mata kuliah Teknik Analog dan Digital. Hal ini menunjukkan bahwa penggunaan GUI

matlan sangat membantu dosen dalam menyampaikan materi *operational amplifier* kepada mahasiswa. Penggunaan Matlab tidak hanya pada *operational amplifier* semata, tetapi dapat dikembangkan pada materi-materi yang berhubungan dengan kendali maupun sinyal.

5. Kesimpulan

Berdasarkan hasil pembahasan yang dilakukan dapat diambil beberapa kesimpulan sebagai berikut:

- a. Pengembangan GUI MATLAB untuk aplikasi Op-Amp sangat membantu mahasiswa dalam memahami materi secara keseluruhan.
- b. Perancangan GUI MATLAB untuk aplikasi Op-Amp sangat memudahkan untuk mendapatkan hasil perhitungan yang tepat.
- c. Penerimaan materi GUI MATLAB untuk aplikasi Op-Amp lebih baik dibandingkan dengan metode pembelajaran konvensional.
- d. Pengembangan metode pembelajaran kreatif, inovatif dan terbaru perlu terus ditingkatkan untuk meningkatkan pemahaman dan prestasi mahasiswa yang lebih baik dalam mempelajari materi.

DAFTAR PUSTAKA

- Anonymous. 2010. *Loctronics Simplifying Electricity Operational Amplifier*. Matrix Multimedia
- Attia, John Okyere. 1999. *Electronics and circuit analysis using MATLAB*. CRC Press.
- Clayton, George and Winder, Steve. 2003. *Operational Amplifiers Fifth Edition*. Newnes.
- Jung, Walt. ed. 2005. *Op Amp Applications Handbook*. Newnes

Lampiran

Perancangan Aplikasi OP-AMP Dengan Software GUI Matlab

1. Inverting Amplifier

kasus 1

```

%masukan
Vin = str2double(get(handles.Vin_1_Edit, 'string'));
R1 = str2double(get(handles.R_1_Edit, 'string'));
R2 = str2double(get(handles.R_2_Edit, 'string'));

%rumus
Vout = (-R2/R1)*Vin;
Gain = (Vin/-Vout);

%hasil
set(handles.Vout_Out_Text, 'String', num2str(Vout));
set(handles.Gain_Out_Text, 'string', num2str(Gain));
%plot

n=1;
t = 0:pi/8: 2*n*pi;
Vin_Plot = Vin * sin(t);
Vout_Plot = Vout * sin(t);
Power_Supply =9;
if (abs (Vout) <= abs(Power_Supply))
    axes(handles.Vin_Vout_Axes)
    plot(t, Vin_Plot, 'RED', 'linewidth', 2);
    grid on;
    axis([ 0 max(t) -30 30]);
    hold on;
    plot(t, Vout_Plot, 'GREEN', 'linewidth', 2);
    grid on;
    xlabel('Time (t)', 'fontweight', 'bold');
    ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
    legend('Vin', 'Vout');
else
    axes(handles.Vin_Vout_Axes);
    Vin_Plot = Vin * sin(t);
    Vout_Plot = Vout * sin(t);
    for i = 1: length(Vout_Plot);
        if (Vout_Plot(i)>Power_Supply);
            Vout_Plot(i)= Power_Supply;
        elseif (Vout_Plot(i) < -Power_Supply);
            Vout_Plot(i)= -Power_Supply;
        end
    end
    plot(t, Vin_Plot, 'RED', 'linewidth', 2);
    grid on;
    axis([ 0 max(t) -30 30]);
    hold on;
    plot(t, Vout_Plot, 'GREEN', 'linewidth', 2);
    grid on;
    xlabel('Time (t)', 'fontweight', 'bold');
    ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
    legend('Vin', 'Vout');
    endxlabel('Time (t)', 'fontweight', 'bold');

```

```

        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    end
    %----- Display warning message (Clipping)-----%
    if abs(Vout) > 16
        msgboxText{1} = 'Clipping!';
        msgbox(msgboxText, 'Clipping Phenomena', 'warn');
    end
end

```

2. Non-Inverting Amplifier

kasus 2

```

%masukan
Vin = str2double(get(handles.Vin_1_Edit, 'string'));
R1 = str2double(get(handles.R_1_Edit, 'string'));
R2 = str2double(get(handles.R_2_Edit, 'string'));

%-----Rumus-----%

Vout = (R2/R1+1)*Vin;
Gain = (Vout/Vin);
    set(handles.Vout_Out_Text, 'String', num2str(Vout));
    set(handles.Gain_Out_Text, 'String', num2str(Gain));

%plot

n=1;
t = 0:pi/8: 2*n*pi;
Vin_Plot = Vin * sin(t);
Vout_Plot = Vout * sin(t);
Power_Supply = 9;
if (abs (Vout) <= abs(Power_Supply))
    axes(handles.Vin_Vout_Axes)
    plot(t, Vin_Plot, 'RED' , 'linewidth', 2);
    grid on;
    axis([ 0 max(t) -30 30]);
    hold on;
    plot(t, Vout_Plot, 'GREEN' , 'linewidth', 2);
    grid on;
    xlabel('Time (t)', 'fontweight', 'bold');
    ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
    legend('Vin', 'Vout');
else
    axes(handles.Vin_Vout_Axes);
    Vin_Plot = Vin * sin(t);
    Vout_Plot = Vout * sin(t);
    for i = 1: length(Vout_Plot);
        if (Vout_Plot(i)>Power_Supply);
            Vout_Plot(i)= Power_Supply;
        elseif (Vout_Plot(i) < -Power_Supply);
            Vout_Plot(i)= -Power_Supply;
        end
    end
    end
    plot(t, Vin_Plot, 'RED' , 'linewidth', 2);
    grid on;
    axis([ 0 max(t) -30 30]);
    hold on;

```

```

        plot(t, Vout_Plot,'GREEN' , 'linewidth',2);
        grid on;
        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
        endxlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    end
    %----- Display warning message (Clipping)-----%
    if abs(Vout) > 16
        msgboxText{1} = 'Clipping!';
        msgbox(msgboxText, 'Clipping Phenomena', 'warn');
    end

kasus 3
    Vin = str2double(get(handles.Vin_1_Edit, 'string'));
    Vout = Vin;
    Gain = Vin/Vout;

    set(handles.Vout_Out_Text, 'string', num2str(Vout));
    set(handles.Gain_Out_Text, 'string', num2str(Gain));

    set(handles.Vout_Out_Text, 'String', num2str(Vout));
    n=1;
    t = 0:pi/8: 2*n*pi;
    Vin_Plot = Vin * sin(t);
    Vout_Plot = Vout * sin(t);
    Power_Supply=9;
    if (abs (Vout) <= abs(Power_Supply))
        axes(handles.Vin_Vout_Axes)
        plot(t, Vin_Plot,'RED' , 'linewidth',2);
        grid on;
        axis([ 0 max(t) -30 30]);
        hold on;
        plot(t, Vout_Plot,'GREEN' , 'linewidth',2);
        grid on;
        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    else
        axes(handles.Vin_Vout_Axes);
        Vin_Plot = Vin * sin(t);
        Vout_Plot = Vout * sin(t);
        for i = 1: length(Vout_Plot);
            if (Vout_Plot(i)>Power_Supply);
                Vout_Plot(i)= Power_Supply;
            elseif (Vout_Plot(i) < -Power_Supply);
                Vout_Plot(i)= -Power_Supply;
            end
        end
        end
        plot(t, Vin_Plot,'RED' , 'linewidth',2);
        grid on;
        axis([ 0 max(t) -30 30]);
        hold on;
        plot(t, Vout_Plot,'GREEN' , 'linewidth',2);
        grid on;
        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
        endxlabel('Time (t)', 'fontweight', 'bold');
    end

```

```

        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    end
    %----- Display warning message (Clipping)-----%
    if abs(Vout) > 16
        msgboxText{1} = 'Clipping!';
        msgbox(msgboxText, 'Clipping Phenomena', 'warn');
    end
end

```

3. Summing Amplifier

kasus 4

```

    %masukan
    Vin = str2double(get(handles.Vin_1_Edit, 'string'));
    Vin2 = str2double(get(handles.Vin_2_Edit, 'string'));
    R1 = str2double(get(handles.R_1_Edit, 'String'));
    R2 = str2double(get(handles.R_2_Edit, 'String'));
    Rf = str2double(get(handles.RF_Edit, 'string'));

    %-----Summing Equation -----%

    %rumus
    Vout = -Rf*(1/R1*Vin+1/R2*Vin2);

    set(handles.Vout_Out_Text, 'String', num2str(Vout));
    n=1;
    t = 0:pi/8: 2*n*pi;
    Vin_Plot = Vin * sin(t);
    Vout_Plot = Vout * sin(t);
    Power_Supply=9;
    if (abs (Vout) <= abs(Power_Supply))
        axes(handles.Vin_Vout_Axes)
        plot(t, Vin_Plot, 'RED' , 'linewidth', 2);
        grid on;
        axis([ 0 max(t) -30 30]);
        hold on;
        plot(t, Vout_Plot, 'GREEN' , 'linewidth', 2);
        grid on;
        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    else
        axes(handles.Vin_Vout_Axes);
        Vin_Plot = Vin * sin(t);
        Vout_Plot = Vout * sin(t);
        for i = 1: length(Vout_Plot);
            if (Vout_Plot(i)>Power_Supply);
                Vout_Plot(i)= Power_Supply;
            elseif (Vout_Plot(i) < -Power_Supply);
                Vout_Plot(i)= -Power_Supply;
            end
        end
        plot(t, Vin_Plot, 'RED' , 'linewidth', 2);
        grid on;
        axis([ 0 max(t) -30 30]);
        hold on;
        plot(t, Vout_Plot, 'GREEN' , 'linewidth', 2);
        grid on;
    end
end

```



```

        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
        endxlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vout');
    end
    %----- Display warning message (Clipping)-----%
    if abs(Vout) > 16
        msgboxText{1} = 'Clipping!';
        msgbox(msgboxText, 'Clipping Phenomena', 'warn');
    end

```

4. Differential Amplifier

kasus 5

```

    %masukan
    Vin = str2double(get(handles.Vin_1_Edit, 'string'));
    Vin2 = str2double(get(handles.Vin_2_Edit, 'string'));
    R1 = str2double(get(handles.R_1_Edit, 'String'));
    R2 = str2double(get(handles.R_2_Edit, 'String'));
    R3 = str2double(get(handles.R_3_Edit, 'string'));
    R4 = str2double(get(handles.RF_Edit, 'string'));

    %rumus
    Vout = (-(R2/R1)*Vin)+(R4/(R3+R4))*(1+(R2/R1))*(Vin2);

    set(handles.Vout_Out_Text, 'String', num2str(Vout));

    n=1;
    t = 0:pi/8: 2*n*pi;
    Vin_Plot = Vin * sin(t);
    Vin2_Plot = Vin2 * sin(t);
    Vout_Plot = Vout * sin(t);
    Power_Supply=9;
    if (abs (Vout) <= abs(Power_Supply))
        axes(handles.Vin_Vout_Axes)
        plot(t, Vin_Plot, 'RED', 'linewidth', 2);
        grid on;
        axis([ 0 max(t) -30 30]);
        hold on;
        plot(t, Vin2_Plot, 'BLUE', 'linewidth', 2);
        grid on;
        hold on;
        plot(t, Vout_Plot, 'GREEN', 'linewidth', 2);
        grid on;
        xlabel('Time (t)', 'fontweight', 'bold');
        ylabel('Input - Output Voltage (V)', 'fontweight', 'bold');
        legend('Vin', 'Vin2', 'Vout');
    else
        axes(handles.Vin_Vout_Axes);
        Vin_Plot = Vin * sin(t);
        Vin2_Plot = Vin2 * sin(t);
        Vout_Plot = Vout * sin(t);
        for i = 1: length(Vout_Plot);
            if (Vout_Plot(i)>Power_Supply);
                Vout_Plot(i)= Power_Supply;
            elseif (Vout_Plot(i) < -Power_Supply);
                Vout_Plot(i)= -Power_Supply;
            end
        end
    end

```

```
end
plot(t, Vin_Plot,'RED' , 'linewidth',2);
grid on;
axis([ 0 max(t) -30 30]);
hold on;
plot(t, Vin2_Plot,'BLUE' , 'linewidth',2);
grid on;
hold on;
plot(t, Vout_Plot,'GREEN' , 'linewidth',2);
grid on;
xlabel('Time (t)','fontweight','bold');
ylabel('Input - Output Voltage (V)','fontweight','bold');
legend('Vin', 'Vin2', 'Vout');
endxlabel('Time (t)','fontweight','bold');
ylabel('Input - Output Voltage (V)','fontweight','bold');
legend('Vin', 'Vout');
end
```