



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202107046, 19 Januari 2021

Pencipta

Nama : Asep Andang, MT, Ir. Rukmi Sari Hartati, MT, PhD dkk

Alamat : Jalan Cinehel No. 29 Cipedes, Tasikmalaya, JAWA BARAT, 46133

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Asep Andang, MT

Alamat : Jalan Cinehel No. 29 Cipedes, Tasikmalaya, JAWA BARAT, 46133

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : Program Reduksi Arus Harmonisa Menggunakan Filter Daya Aktif Hibrid Dengan Kendali Finite Model Predictive Control Pada Jaringan Listrik Menggunakan Matlab-Simulink

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 19 Januari 2021, di Denpasar

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000234502

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Asep Andang, MT	Jalan Cinchel No. 29 Cipedes
2	Ir. Rukmi Sari Hartati, MT, PhD	Jalan Kertha Petasikan I No. 4
3	Ida Bagus Gede Manuaba, ST, MT	Jalan Padang Galak No. 18
4	I Nyoman Satya Kumara	Jalan Batur No. 10A



**Program Reduksi Arus Harmonisa menggunakan Filter Daya Aktif Hibrid
dengan kendali Finite Model Predictive Control pada jaringan listrik
menggunakan Matlab-Simulink**

Oleh:

Asep Andang

Rukmi Sari Hartati

Ida Gede bagus Manuaba

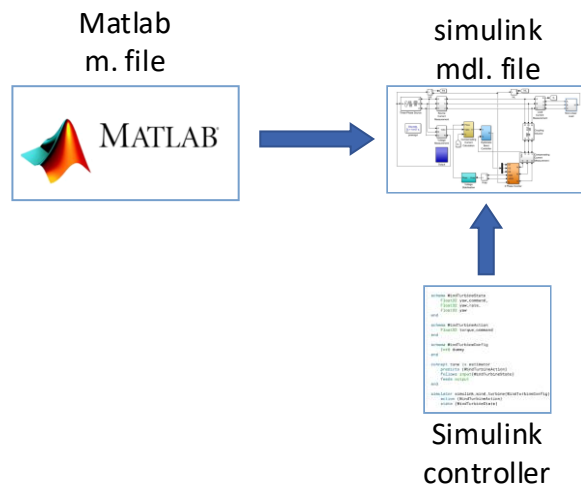
I nyoman Satya Kumara

1. Uraian Singkat

Program ini berfungsi untuk mereduksi harmonisa pada jaringan distribusi listrik dengan beban nonlinier yang direpresentasikan sebagai penyearah satu fasa terhubung dengan RL. Sistem reduksi harmonisa ini menggunakan filter LC sebagai filter pasif dan grid connected inverter 4 kawat. Kendali untuk pengsaklaran inverter ini menggunakan metode Model Predictive Control dengan set model adalah switching state sejumlah 2^4 kondisi atau 16 kondisi pengsaklaran. MPC ini mengambil data dari arus beban, tegangan sumber serta arus filter sebelumnya kemudian di proses dengan ekstrapolasi dan proses prediksi dengan menggunakan fungsi biaya untuk menentukan kondisi pengsaklaran optimal.

2. Hierarki Program

Untuk menjelaskan hierarki program ini dapat dilihat pada gambar 1, yang menjelaskan hubungan antar file program dalam matlab dan simulink



Gambar 1. Hierarki Program

Program Reduksi Arus Harmonisa menggunakan Filter Daya Aktif Hibrid dengan kendali Finite Model Predictive Control pada jaringan listrik menggunakan Matlab-Simulink ini terdiri dari file matlab yaitu **mpc_param_SAPF_MPC_3ph_4wire_LC_2015aa.m** dan file simulink **SAPF_MPC_3ph_4wire_LC_seri_17a.slx**

m.file matlab yang berisi parameter beban pasif dan kondisi pasif dari sistem, sementara simulink berisi rangkaian yang akan disimulasikan dengan parameter diambil dari m.file, dalam simulink terdapat blok program yang mengendalikan sistem dalam simulink yang menggunakan syntax matlab yaitu simulink controller. Ketiga bagian ini yang bekerja untuk mensimulasikan program reduksi harmonisa ini.

3. Tampilan program m.file

```
Ts = 10e-6;
Vs = 400;
Vdc = 725;
Ls = 0.5e-3;
Rs = 0.001;
Lf = 4e-3;
%[Lfa, Lfb, Lfc, Lfn, Ln] = deal(Lf);
Cf = 500e-6;
Rf = 0.022;
%[Rfa, Rfb, Rfc, Rfn, Rn] = deal(Rf);
```

```

Cdc = 2350e-6;
%Leq = inv(1/Lfa + 1/Lfb + 1/Lfc + 1/Lfn);
periode = 20e-3;
RL = 20;
LL = 0.07;

% switching tiap kondisi
S(:, :, 1) = [1;0;0;0];
S(:, :, 2) = [1;1;0;0];
S(:, :, 3) = [0;1;0;0];
S(:, :, 4) = [0;1;1;0];
S(:, :, 5) = [0;0;1;0];
S(:, :, 6) = [1;0;1;0];
S(:, :, 7) = [1;1;1;0];
S(:, :, 8) = [0;0;0;0];
S(:, :, 9) = [1;0;0;1];
S(:, :, 10) = [1;1;0;1];
S(:, :, 11) = [0;1;0;1];
S(:, :, 12) = [0;1;1;1];
S(:, :, 13) = [0;0;1;1];
S(:, :, 14) = [1;0;1;1];
S(:, :, 15) = [1;1;1;1];
S(:, :, 16) = [0;0;0;1];

for i = 1:16
    w(i)=0;
end
for i = 1:16
    pht(i, :)= [0 0 0];
end

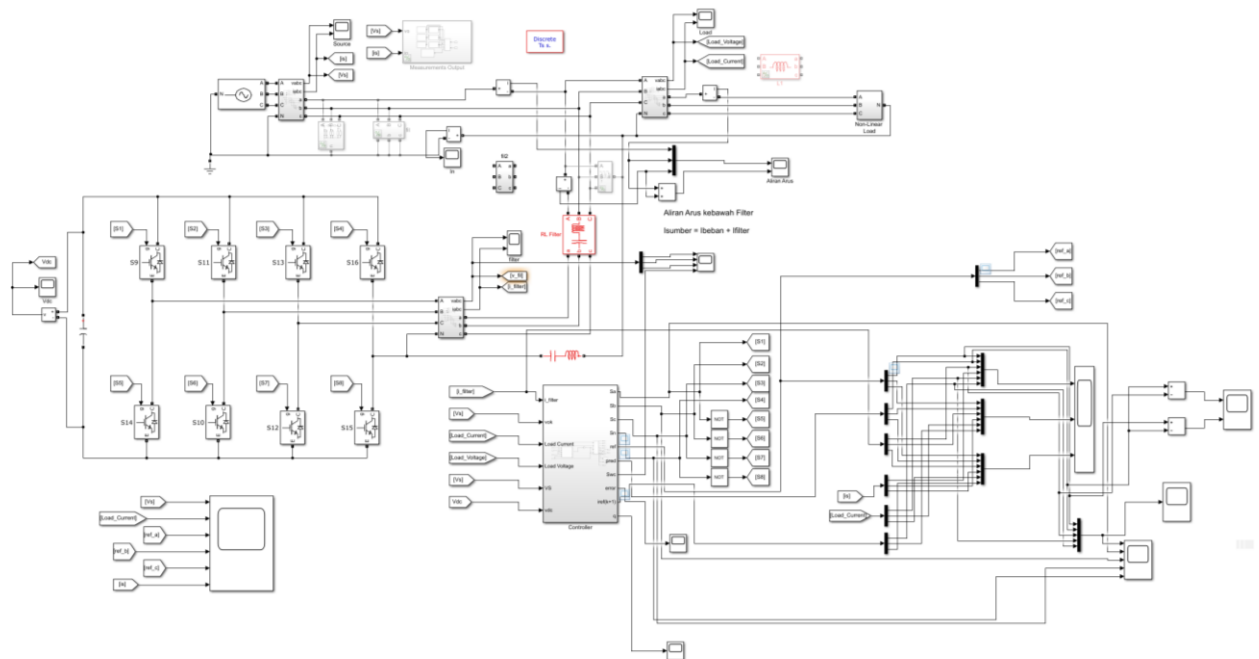
```

pada m.file ini berisi Time sampling yaitu 10 μ s, kemudian nilai parameter Vs, Vdc, Ls, Rs, Lf, Cf, Rf, dan Cdc kemudian beban awal RL dan LL

pada m.file ini juga terdapat switching state untuk 16 kondisi dari 4 kaki inverter dalam bentuk array matriks. M.file ini akan bekerja pada file simulink sebagai kondisi awal.

4. Tampilan simulink

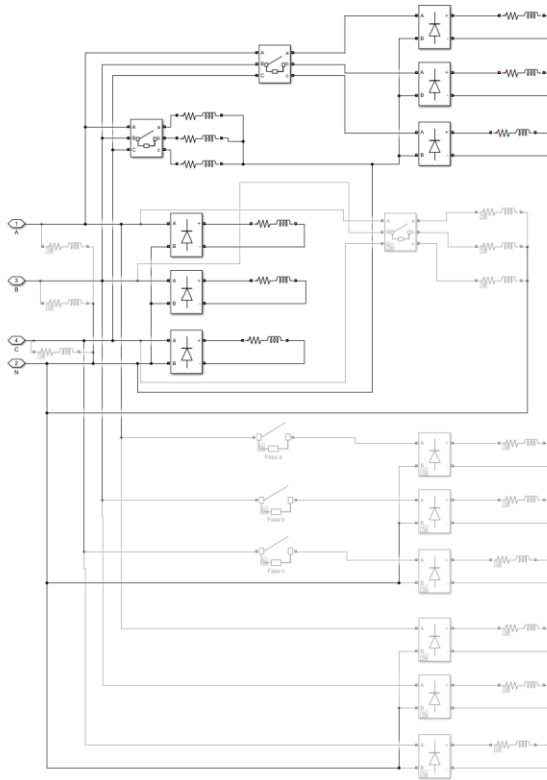
File simulink berupa rangkaian yang terhubung dengan m.file dengan menggunakan callback dalam initialfunction sehingga file simulink ini terkunci dengan program m.file tertentu. Adapun gambar rangkaian utama dari simulink ini seperti di bawah ini



Gambar 2. Rangkaian utama

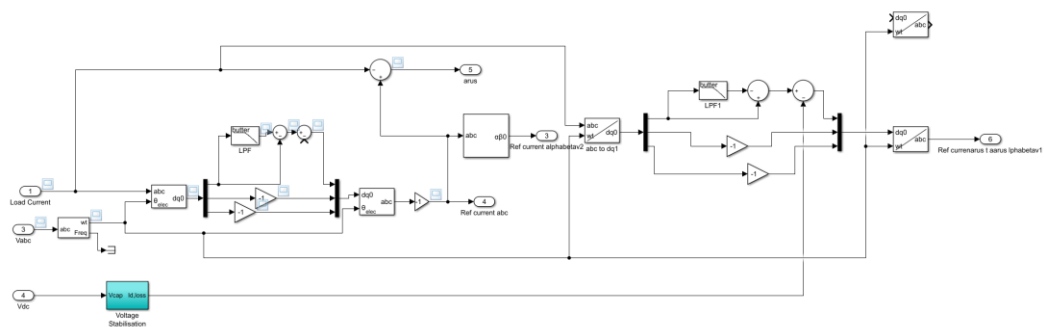
Gambar 2 memperlihatkan rangkaian sumber berupa Vs yang terhubung dengan beban yaitu blok nonlinier load yang dijelaskan dalam gambar 3. Beban ini terhubung paralel dengan shunt hybrid active power filter dengan filter LC sebagai filter daya pasif dan inverter sebagai filter daya aktif dimana kendali MPC terhubung dalam blok kendali MPC yang akan di jelaskan dalam gambar 4.

Beban nonlinier yang terhubung dalam rangkaian utama dapat dilihat pada gambar 4 yang berupa subsistem.



Gambar 3. Beban nonlinier

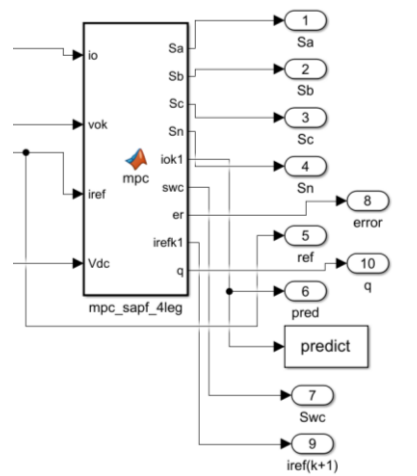
Dalam beban ini terdapat tiga buah penyearah satu fasa yang terhubung dengan beban Resistor, induktor dan netral berupa RL dan LL dengan parameter yang terdapat pada m.file, kemudian beban lain berupa beban linier Resistor dan induktor lainnya tiap fasa yang terhubung secara Y dengan empat kawat, beban lain adalah beban dinamis dimana beban ini terhubung dengan saklar yang di beri waktu tertentu untuk mengetahui sejauh mana proses kestabilan program dalam mereduksi harmonisa.



Gambar 4. Pembangkitan arus referensi

Sistem pengendalian inverter terdapat dalam gambar 4 yang memperlihatkan blok pembangkitan arus referensi yang mempunyai masukan dari tegangan dc V_{dc} , arus beban I_L dan tegangan sumber V_s dengan transformasi park maka dihasilkan i_d , i_q dan i_0 . Kemudian i_d dilewatkan pada blok low pass filter dengan tegangan cuttof 20 Hz sehingga akan menghasilkan

arus I_d (dc) dan dikurangkan i_d sehingga menghasilkan arus i_d (ac) dan ditambah dengan i_{dc} yang berasal dari regulasi tegangan dc sehingga menghasilkan arus i_{dref} setelah melalui proses re-transformasi park sehingga akan menghasilkan arus referensi untuk kendali MPC berupa I^*_{Labc} yang akan terhubung dengan blok MPC pada gambar 5.



Gambar 5. Kendali MPC

Gambar 5 memperlihatkan blok kendali MPC yang mempunyai masukan yaitu arus referensi I_{ref} , tegangan dc v_{dc} , tegangan sumber V_{oc} dan arus sumber i_o , dalam blok ini diuraikan cara kerja sistem simulink seperti pada bagian 4.

5. Tampilan program simulink

Blok MPC dalam simulink diatas bila di uraikan seperti pada list program `controller/mpc_shafp_4leg` di bawah ini

```
function [Sa,Sb,Sc,Sn,iok1] = mpc(io, vok, iref,Vdc, Ts, Rf, Lf, S, Cf, Ls)

% menyimpan nilai
persistent iok1 prediksi_ke irefk_3 irefk_2 irefk_1

% inisialisasi variabel
if isempty(iok_1)
    iok_1 = [0;0;0];
end
if isempty(irefk_1)
    irefk_1 = [0;0;0];
end
if isempty(irefk_2)
    irefk_2 = [0;0;0];
end
if isempty(irefk_3)
    irefk_3 = [0;0;0];
end
if isempty(prediksi_ke)
    prediksi_ke = 0;
end

wopt = 1e10;
sw = 0;

irefk1 = 4*iref - 6*irefk_1 + 4*irefk_2 - irefk_3;
irefk_3 = irefk_2;
irefk_2 = irefk_1;
irefk_1 = iref;

prediksi_ke = prediksi_ke + 1;
```



```

for i = 1:16
    v_state = [1 0 0 -1; 0 1 0 -1; 0 0 1 -1]*S(:,i)*Vdc;
    iokl = (1/((Lf+Ls)/Ts+Rf+Ts/Cf))*(-vok + ((Lf+Ls)/Ts)*io - (Ts/Cf)*io + v_state);
    w = (irefk1(1)-iokl(1))^2+(irefk1(2)-iokl(2))^2+(irefk1(3)-iokl(3))^2;
    if (w<wopt)
        wopt = w;
        sw = i;
    end
end

% menerapkan switch yang dipilih
Sa = S(1,sw);
Sb = S(2,sw);
Sc = S(3,sw);
Sn = S(4,sw);
swc = sw;
snk = Sn;

```

6. Penjelasan Controller Simulink

```
function [Sa,Sb,Sc,Sn,iokl] = mpc(io, vok, iref, Vdc, Ts, Rf, Lf, S, Cf)
```

Bagian [Sa,Sb,Sc,Sn,iokl] merupakan output dari blok MPC yang digunakan untuk mengendalikan kaki dari konverter kecuali iokl yang hanya digunakan untuk melihat/memploting arus hasil prediksi. Keluaran Sa, Sb, Sc, Sn hanya bernilai 0 atau 1, sedangkan iokl adalah matrik 3x1 sesuai urutan fasa a fasa b fasa c.

Bagian io, vok, iref, Vdc merupakan input dari blok MPC, menyalurkan nilai hasil pengukuran kedalam blok MPC yang digunakan untuk menghitung hasil prediksi.

Bagian Ts, Rf, Lf, S, Cf merupakan parameter yang diambil dari **base workspace matlab**.

Untuk melihat input-output-parameter dapat dilihat di **data manager**

```

% menyimpan nilai
persistent iokl prediksi_ke irefk_3 irefk_2 irefk_1

```

fungsi persistent adalah untuk mempertahankan/menyimpan nilai dari variabel iokl, prediksi_ke, irefk_3, irefk_2 dan irefk_1 kedalam memori, sehingga dapat digunakan diperhitungan selanjutnya.

iokl = io(k+1) sebagai arus prediksi $i_o^p(k+1)$

irefk_3 = iref(k-3) sebagai arus referensi k-3

irefk_2 = iref(k-2) sebagai arus referensi k-2

irefk_1 = iref(k-1) sebagai arus referensi k-1

```

if isempty(iok_1)
    iok_1 = [0;0;0];
end
if isempty(irefk_1)
    irefk_1 = [0;0;0];
end
if isempty(irefk_2)
    irefk_2 = [0;0;0];
end
if isempty(irefk_3)
    irefk_3 = [0;0;0];
end
if isempty(prediksi_ke)
    prediksi_ke = 0;
end

```

jika variabel iok_1 ($i_o(k-1)$) kosong, maka iok_1 diberi nilai awal $[0;0;0]$, sehingga i_o menjadi:

$$i_o(k-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

karena iok_1 variabel yang mengandung nilai fasa a, fasa b, fasa c maka diberi awalan dengan nilai matrik 3x1. Hal ini dibutuhkan karena ketika pertamakali simulasi dijalankan (di *run*) variabel tersebut kosong, tidak bernilai. Ketika program berjalan, nilai variabel iok_1 diupdate dan nilai terakhir disimpan sehingga variabel iok_1 tidak lagi kosong sehingga fungsi "if" tidak terpanggil.

Jika variabel iok_1 tidak kosong, maka nilai iok_1 diambil dari nilai sebelumnya yang telah disimpan di memori dengan fungsi `persistent` diatas.

Demikian pula untuk $irefk_1$, $irefk_2$ dan $irefk_3$ sehingga nilainya menjadi

$$i_{ref}(k-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, i_{ref}(k-2) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, i_{ref}(k-3) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```

wopt = 1e10;
sw = 0;

```

Inisialisasi variabel cost function $wopt$ dengan kondisi awal bernilai sangat besar dan variabel switching state sw dengan kondisi awal switching 0.

```

% ekstrapolasi referensi
irefk1 = 4*iref - 6*irefk_1 + 4*irefk_2 - irefk_3;

```

Bila di tulis ulang maka persamaan diatas akan menjadi

$$i_{ref}(k+1) = 4i_{ref} - 6i_{ref}(k-1) + 4i_{ref}(k-2) - i_{ref}(k-3)$$

$irefk_1$ merupakan (k-1) nilai, $irefk_2$ merupakan (k-2), $irefk_3$ merupakan (k-3), nilai ini didapat dari memori yang disimpan dengan fungsi `persistent`.

```

irefk_3 = irefk_2;
irefk_2 = irefk_1;
irefk_1 = iref;

```

```

 $i_{ref}(k-3) = i_{ref}(k-2);$ 
 $i_{ref}(k-2) = i_{ref}(k-1);$ 
 $i_{ref}(k-1) = i_{ref};$ 

```

Fungsi diatas berguna untuk mengupdate variabel dari i_{refk_3} menjadi i_{refk_2} , kemudian i_{refk_2} menjadi i_{refk_1} , dan i_{refk_1} menjadi i_{ref} . Hal ini dibutuhkan untuk menghitung ekstrapolasi selanjutnya.

```
for i = 1:16
```

Fungsi “for” di atas untuk menghitung variabel counter “i” (bukan arus). i = 1,2,3,4 hingga bernilai 16. Merepresentasikan switching state yang berjumlah 16 buah sehingga satu siklus for loop (i = 1) adalah satu switching state, kemudian satu prediksi dan didapatkan satu cost function. Setelah selesai satu siklus variabel “i” di counter (i = 2) dan merepresentasikan switching kedua, prediksi kedua dan cost function kedua. Hal ini terus dilakukan hingga “i” sama dengan 16.

```
v_state = [1 0 0 -1; 0 1 0 -1; 0 0 1 -1]*S(:, :, i)*Vdc;
```

$$v_{state} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} * S(:, :, i) * V_{dc}$$

dimana $S(:, :, i)$ ada pada m-file sistem seperti di bawah ini

```

% switching tiap kondisi
S(:, :, 1) = [1;0;0;0];
S(:, :, 2) = [1;1;0;0];
S(:, :, 3) = [0;1;0;0];
S(:, :, 4) = [0;1;1;0];
S(:, :, 5) = [0;0;1;0];
S(:, :, 6) = [1;0;1;0];
S(:, :, 7) = [1;1;1;0];
S(:, :, 8) = [0;0;0;0];
S(:, :, 9) = [1;0;0;1];
S(:, :, 10) = [1;1;0;1];
S(:, :, 11) = [0;1;0;1];
S(:, :, 12) = [0;1;1;1];
S(:, :, 13) = [0;0;1;1];
S(:, :, 14) = [1;0;1;1];
S(:, :, 15) = [1;1;1;1];
S(:, :, 16) = [0;0;0;1];

```

contoh v_{state} untuk $i=1$ maka $S(:, :, 1) = [1;0;0;0]$ sehingga

$$v_{state} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} * V_{dc}$$

$$v_{state} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} * V_{dc} = \begin{bmatrix} V_{dc} \\ 0 \\ 0 \end{bmatrix}$$

untuk v_{state} dengan $i=5$ misalnya maka $S(:, :, 5) = [0;0;1;0]$ sehingga

$$v_{state} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} * [0 \quad 0 \quad 1 \quad 0] * V_{dc}$$

$$v_{state} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} * V_{dc} = \begin{bmatrix} 0 \\ 0 \\ V_{dc} \end{bmatrix}$$

atau untuk $i = 12$ maka $s(:, :, 12) = [0; 1; 1; 1]$ sehingga

$$v_{state} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} * [0 \quad 1 \quad 1 \quad 1] * V_{dc}$$

$$v_{state} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} * V_{dc} = \begin{bmatrix} -V_{dc} \\ 0 \\ 0 \end{bmatrix}$$

Persamaan ini berfungsi untuk mengkonversi switching state dari bentuk $(1 \ 0 \ 0 \ 0, \ 1 \ 1 \ 0 \ 0, \dots, \ 0 \ 0 \ 0 \ 1)$ menjadi bentuk tegangan $(V_{dc} \ 0 \ 0, \ V_{dc} \ V_{dc} \ 0, \dots, \ -V_{dc} \ -V_{dc} \ -V_{dc})$.

$$i_{ok1} = (1 / (L_f / T_s + R_f + T_s / C_f)) * (-v_{ok} + (L_f / T_s) * i_o - (T_s / C_f) * i_o + v_{state});$$

$$i_o^{pre}[k+1] = \frac{1}{\frac{L_f}{T_s} + R_f + \frac{T_s}{C_f}} x \left(-v_o[k] + i_o[k] \left(\frac{L_f}{T_s} - \frac{T_s}{C_f} \right) + v_{state} \right)$$

Merupakan persamaan arus prediksi, pada persamaan diatas i_o , v_{ok} , dan v_{state} adalah matriks 3x1 merepresentasikan fasa a,b,c sehingga hasil i_{ok1} juga merupakan matriks 3x1 merepresentasikan arus prediksi tiap fasa a,b,c.

$$w = (i_{refk1}(1) - i_{ok1}(1))^2 + (i_{refk1}(2) - i_{ok1}(2))^2 + (i_{refk1}(3) - i_{ok1}(3))^2;$$

$$w = [i_{ref}(k+1)(1) - i_o(k+1)(1)]^2 + [i_{ref}(k+1)(2) - i_o(k+1)(2)]^2 + [i_{ref}(k+1)(3) - i_o(k+1)(3)]^2$$

Merupakan cost function, $i_{refk1}(1)$ merepresentasikan nilai i_{refk1} pada matriks pertama, $i_{refk1}(2)$ pada matriks kedua, $i_{refk1}(3)$ pada matriks ketiga. Sama seperti i_{refk1} , pada i_{ok1} pun dilakukan hal yang sama, ini bertujuan agar referensi fasa a dikurangi arus prediksi fasa a.

```
if (w < wopt)
    wopt = w;
    sw = i;
end
```

Jadi bila error yang paling kecil terjadi pada $i = 8$ maka nilai $sw = 8$

Digunakan untuk meng-overwrite cost function dan switching state ke-i jika cost function yang sekarang ternyata lebih kecil dibanding cost function yang disimpan sebelumnya.

```
Sa = S(1, sw);  
Sb = S(2, sw);  
Sc = S(3, sw);  
Sn = S(4, sw);
```

Menerapkan switching yang paling optimal pada variabel sw ($sw = 1, 2, 3, \dots, 16$) lalu memisahkan tiap kakinya disesuaikan dengan fungsi $S(x, sw)$, ($x = 1, 2, 3, 4$) merepresentasikan a,b,c,n. lihat tabel switching contohnya bila nilai $sw = 12$ maka

nilai Sa = nilai kolom 1 dengan $sw = 12$ yaitu $sw = 0$

nilai Sb = nilai kolom 2 dengan $sw = 12$ yaitu $sw = 1$

nilai Sb = nilai kolom 3 dengan $sw = 12$ yaitu $sw = 1$

nilai Sb = nilai kolom 4 dengan $sw = 12$ yaitu $sw = 1$

sehingga pengsaklaran inverter akan mempunyai nilai $[0; 1; 1; 1]$