I. Source code for Fuzzy Interface Service:

```
function [avg, diff] = fcn(SOC1,SOC2,SOC3,SOC4)
High = max([SOC1 SOC2 SOC3 SOC4]);
Low = min([SOC1 SOC2 SOC3 SOC4]);
if High==SOC1&&Low==SOC2
avg=(SOC1+SOC2)*0.5;
diff=SOC1-SOC2;
elseif High==SOC1&&Low==SOC3
avg=(SOC1+SOC3)*0.5;
diff=SOC1-SOC3;
elseif High==SOC1&&Low==SOC4
avg=(SOC1+SOC4)*0.5;
diff=SOC1-SOC4;
elseif High==SOC2&&Low==SOC1
avg=(SOC2+SOC1)*0.5;
diff=SOC2-SOC1;
elseif High==SOC2&&Low==SOC3
avg=(SOC2+SOC3)*0.5;
diff=SOC2-SOC3;
elseif High==SOC2&&Low==SOC4
avg=(SOC2+SOC4)*0.5;
diff=SOC2-SOC4;
elseif High==SOC3&&Low==SOC1
avg=(SOC3+SOC1)*0.5;
diff=SOC3-SOC1;
elseif High==SOC3&&Low==SOC2
avg=(SOC3+SOC2)*0.5;
diff=SOC3-SOC2;
elseif High==SOC3&&Low==SOC4
```

```
avg=(SOC3+SOC4)*0.5;
diff=SOC3-SOC4;
elseif High==SOC4&&Low==SOC1
avg=(SOC4+SOC1)*0.5;
diff=SOC4-SOC1;
elseif High==SOC4&&Low==SOC2
avg=(SOC4+SOC2)*0.5;
diff=SOC4-SOC2;
elseif High==SOC4&&Low==SOC3
avg=(SOC4+SOC3)*0.5;
diff=SOC4-SOC3;
else
avg=0;
diff=0;
end
end
```

II. Source code for Balancing Control

```
function [s1,s2,s3,s4,s5,s6,s7,s8] = fcn(SOC1,SOC2,SOC3,SOC4,PWM1,PWM2)
High = max([SOC1 SOC2 SOC3 SOC4]);
Low = min([SOC1 SOC2 SOC3 SOC4]);
if High==SOC1&&Low==SOC2
s1=PWM1; s2=1; s5=PWM2;
s3=0; s4=0; s6=0; s7=0; s8=0;
elseif High==SOC1&&Low==SOC3
s1=PWM1; s2=PWM1; s4=PWM2; s7=PWM2;
s3=0; s5=0; s6=0; s8=0;
elseif High==SOC1&&Low==SOC4
s1=PWM1; s2=PWM1; s6=PWM2;
s3=0; s4=0; s5=0; s7=0; s8=0;
```

```
elseif High==SOC2&&Low==SOC1
s3=1; s4=PWM1;
s1=0; s2=0; s5=0; s6=0; s7=0; s8=0;
elseif High==SOC2&&Low==SOC3
s3=PWM1; s4=1; s7=PWM2;
s1=0; s2=0; s5=0; s6=0; s8=0;
elseif High==SOC2&&Low==SOC4
s3=PWM1; s4=PWM1; s6=PWM2;
s1=0; s2=0; s5=0; s7=0; s8=0;
elseif High==SOC3&&Low==SOC1
s3=PWM2; s5=PWM1; s6=PWM1;
s1=0; s2=0; s4=0; s7=0; s8=0;
elseif High==SOC3&&Low==SOC2
s2=PWM2; s5=1; s6=PWM1;
s1=0; s3=0; s4=0; s7=0; s8=0;
elseif High==SOC3&&Low==SOC4
s5=PWM1; s6=1;
s1=0; s2=0; s3=0; s4=0; s7=0; s8=0;
elseif High==SOC4&&Low==SOC1
s3=PWM2; s7=PWM1; s8=PWM1;
s1=0; s2=0; s4=0; s5=0; s6=0;
elseif High==SOC4&&Low==SOC2
s2=PWM2; s5=PWM2; s7=PWM1; s8=PWM1;
s1=0; s3=0; s4=0; s6=0;
elseif High==SOC4&&Low==SOC3
s4=PWM2; s8=PWM1; s7=1;
s1=0; s2=0; s3=0; s5=0; s6=0;
else
s1=0; s2=0; s3=0; s4=0; s5=0; s6=0; s7=0; s8=0;
```

```
end
end
```

III. Source code of Charging

```
function [cv, cc, cu] = fcn(SOC1, SOC2, SOC3, SOC4, ch, dis)
cv=0;
cc=0;
cu=0;
socavg=(SOC1+SOC2+SOC3+SOC4)*0.25;
if ((ch==1) && (dis==0))
if(socavg<80)
cu=10;
cc=1;
cv=0;
elseif(socavg>=80)
cu=0;
cc=0;
cv=1;
else(socavg==100)
cu=0;
cc=0;
cv=0;
end
elseif ((ch==0) && (dis==1))
cu=0;
cc=0;
cv=0;
end
```

Parameter of circuit	Parameter value
Battery nominal Voltage	3.7 V
Battery capacity	3 Ah
Balancing Inductor	1.5 mH
Charging Current	10 A
Charging Voltage	18 V
Battery internal resistance	0.012 ohm
Switching Frequency	5000 HZ
Fuzzy Refresh rate	0.02 s
Initial SoC of Battery 1	70 %
Initial SoC of Battery 2	68 %
Initial SoC of Battery 3	66 %
Initial SoC of Battery 4	64 %