

ภาคผนวก

ภาคผนวก ก. : โปรแกรมการทดสอบคุณสมบัติที่เป็นธรรมชาติของดีซีมอเตอร์

โปรแกรมการทดสอบคุณสมบัติที่เป็นธรรมชาติของดีซีมอเตอร์

```

PROGRAM RK4I.PAS;
(*A PROGRAM FOR SOLVING ORDINARY DIFFERENTIAL.*)
(*EQUATION BY FOURTH-ORDER RUNGE-KUTTA METHOD
*)
(*.READ INITIAL CONITION, NUMBER OF STEPS, *)
(*.AND STEP SIZE:*)
CONST
    TL0 = 0.0;
VAR I,N:INTEGER;
    T,H:REAL; {T=TIME, H=STEP SIZE OF CALULATION}
    K1,K2,K3,K4,L1,L2,L3,L4,M1,M2,M3,M4:REAL; {VARIABLE
OF FOURTH-ORDER
                                RUNGE-KUTTA METHOD}
    X1,X2,X3,TL:REAL; { X1=ARMATURE CURRENT
                                X2=ANGULAR VELOCITY
                                X3=ANGULAR
DISPLACEMENT
                                TL=LOAD TORQUE }
(*-----*)
    FILVAR :TEXT;
    FILENAME :STRING[14];
(*----- FIRST STATE -----*)
FUNCTION FUNCX1D(T,X1,X2,X3:REAL):REAL;
BEGIN
    FUNCX1D:=75.*X1-68.75*X2+125*55;
END;
(*----- SECOND STATE -----*)
FUNCTION FUNCX2D(T,X1,X2,X3,TL:REAL):REAL;
BEGIN
    FUNCX2D:=11.827957*X1-0.086022*X2-21.505376*TL;
END;
(*----- THIRD STATE -----*)
FUNCTION FUNCX3D(T,X1,X2,X3:REAL):REAL;
BEGIN
    FUNCX3D:=X2;
END;
(*-----*)
BEGIN

```

```

    WRITE('ENTER NAME OF FILE TO WRITE:');READLN
(FILENAME);
    ASSIGN(FILVAR,FILENAME); {ASSIGN FILENAME}
    REWRITE(FILVAR); {CONSTRUCT AND OPEN FILE}
    WRITE('ENTER NUMBER OF STEP, STEP SIZE:');READ(N,H);
    WRITE('ENTER INITIAL T,X1,X2,X3:');READ(T,X1,X2,X3);
    WRITELN(FILVAR,'T    X1    X2    X3');
    WRITELN(FILVAR,T:11:5,X1:11:5,X2:11:5,X3:11:5);
    FOR I:=1 TO N DO
        BEGIN
            (*-----*)
            K1:=H*FUNCX1D(T,X1,X2,X3);
            L1:=H*FUNCX2D(T,X1,X2,X3,TL);
            M1:=H*FUNCX3D(T,X1,X2,X3);
            K2:=H*FUNCX1D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1);
            L2:=H*FUNCX2D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1,TL);
            M2:=H*FUNCX3D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1);
            K3:=H*FUNCX1D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2);
            L3:=H*FUNCX2D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2,TL);
            M3:=H*FUNCX3D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2);
            K4:=H*FUNCX1D(T+H,X1+K3,X2+L3,X3+M3);
            L4:=H*FUNCX2D(T+H,X1+K3,X2+L3,X3+M3,TL);
            M4:=H*FUNCX3D(T+H,X1+K3,X2+L3,X3+M3);
            X1:=X1+(K1+2*K2+2*K3+K4)/6;
            X2:=X2+(L1+2*L2+2*L3+L4)/6;
            X3:=X3+(M1+2*M2+2*M3+M4)/6;
            IF I>=1500 THEN TL:=8.
            ELSE TL:=TL0;
            (*-----*)
            T:=T+H;
            WRITELN(FILVAR,T:11:5,X1:11:5,X2:11:5,X3:11:5);
        END;
    END.

```

ภาคผนวก ข. : ค่าพารามิเตอร์ต่างๆของระบบที่ใช้ในการจำลองผล

ตารางที่ ข.1 ค่าพารามิเตอร์ต่างๆของมอเตอร์

ค่าพารามิเตอร์ของระบบที่ใช้ในการจำลองการทำงาน คัดมาจากบทความของ Sousa และ Bose (1994) และ Sen (1981) มีดังต่อไปนี้
- ดีไซน์มอเตอร์แบบกระตุ้นฟลักซ์แยกส่วนขนาดพิกัดแรงดัน 110 V 2.5 hp 1800 rpm
- $R_a = 0.6 \Omega$
- $L_a = 8 \text{ mH}$
- $J = 0.0465 \text{ kg-m}^2$
- $B = 0.004 \text{ N-m.sec/rad}$
- $K_b = 0.55 \text{ V.sec/rad}$
- $I_a = 20 \text{ A}$ กระแสพิกัดของมอเตอร์ (rated motor current)
จากค่าพารามิเตอร์ที่ได้มาสามารถหาค่าอื่นๆ ได้ดังนี้
- มอเตอร์ขนาด 2.5 แรงม้าให้กำลัง เท่ากับ $2.5 \times 746 = 1865 \text{ W}$. (rated power)
- ความเร็ว 1800 rpm เท่ากับ $1800 \times 2\pi/60 = 188.49 \text{ rad/sec}$ (rated speed)
- จาก $P_d = T_d \omega$ ดังนั้น $T_d (\text{rated torque}) = P_d (\text{rated power}) / \omega (\text{rated speed})$ $= 1865 / 188.49 = 9.89 \text{ N-m}$. หมายความว่ามอเตอร์สามารถสร้างแรงบิดเต็มพิกัดได้ 9.89 N-m.
- ค่าคงที่แรงดันย้อนกลับ (back emf constant: K_b) มีความสัมพันธ์กับค่าคงที่แรงบิด (torque constant: K_t) คือ $K_t (\text{N-m./A}) = K_b (\text{V/rad/sec}) = 0.55 \text{ N-m./A}$

ตารางที่ ข.2 ค่าพารามิเตอร์ต่างๆของมอเตอร์ โหลดและตัวควบคุมแบบพีไอ

ค่าพารามิเตอร์และค่าคงที่ของระบบควบคุมวงปิดสำหรับการขับมอเตอร์แบบกระตุ้นฟลักซ์แยกส่วนที่ใช้ในการจำลองการทำงานนำมาจากหนังสือของ Sen (1981) มอเตอร์ที่ใช้เป็นตัวขับ (drive motor) และใช้เป็นโหลด (loading) เป็นมอเตอร์ขนาดเดียวกัน คือ DC motor 110 V, 2.5 hp, 1800 rpm
- $R_A = 1.0 \Omega$; เป็นค่าความต้านทานของวงจรอาร์เมเจอร์รวมกับค่าความต้านทานของตัวตรวจวัดกระแส 0.4Ω ที่ต่ออนุกรมกับมอเตอร์ที่ใช้ขับ
- $L_A = 46 \text{ mH}$; เป็นค่าความเหนี่ยวนำของวงจรอาร์เมเจอร์รวมกับค่าความเหนี่ยวนำภายนอก 40 mH ที่ต่ออนุกรมกับมอเตอร์ที่ใช้ขับ
- $J = 0.093 \text{ kg-m}^2$; สำหรับการต่อเครื่องกลสองตัวที่ใช้เป็นตัวขับและใช้เป็นโหลดเข้าด้วยกัน
- $B = 0.008$; สำหรับการต่อเครื่องกลสองตัวเข้าด้วยกัน
- $I_a = 20 \text{ A}$; กระแสฟลักซ์ของมอเตอร์
- $k_a \Phi = 0.55 \text{ V.sec/rad}$; ค่าคงที่แรงดันเหนี่ยวนำย้อนกลับและค่าคงที่แรงบิด (N-m./A)
- $\tau_a = L_a/R_a = 46 \text{ msec}$; ค่าคงที่ทางเวลาของมอเตอร์เชิงไฟฟ้า
- $\tau_m = J/B = 11.63 \text{ sec}$; ค่าคงที่ทางเวลาของมอเตอร์เชิงกล
- $\tau_t = 0.1 \text{ sec}$; ค่าคงที่ทางเวลาของเครื่องวัดความเร็วรอบของมอเตอร์
- $k_t = 0.057 \text{ V.sec/rad}$; อัตราขยายของเครื่องวัดความเร็วรอบของมอเตอร์
- $k_r = 0.5 \text{ V/A}$; อัตราขยายของตัวตรวจวัดกระแสอาร์เมเจอร์
- $f_c = 200 \text{ Hz}$; ความถี่ในการสับไฟฟ้าของวงจรสับไฟฟ้า
ค่าพารามิเตอร์ของตัวควบคุมแบบพีไอที่ได้จากวิธีที่เหมาะสมที่สุดแบบสมมาตร
- ตัวควบคุมกระแสใช้ค่าพารามิเตอร์ $K_p = 0.129, K_i = 104.16$
- ตัวควบคุมความเร็วใช้ค่าพารามิเตอร์ $K_p = 115.37, K_i = 19.841$
ค่าพารามิเตอร์ของตัวควบคุมแบบพีไอที่ได้จากการทดลองเปลี่ยนค่า (รายละเอียดขั้นตอนการหาค่าพารามิเตอร์อยู่นำถัดไป)
- ตัวควบคุมกระแสใช้ค่าพารามิเตอร์ $K_p = 1.0, K_i = 231$
- ตัวควบคุมความเร็วใช้ค่าพารามิเตอร์ $K_p = 15, K_i = 150$

การหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอ

จากแผนภาพการควบคุมระบบขับเคลื่อนโดยตัวควบคุมแบบพีไอที่ได้แสดงในรูปที่ 3.12 ขั้นตอนการหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอ เพื่อที่จะให้ระบบตอบสนองได้เร็วที่สุดและให้เปอร์เซ็นต์การฟุ้งเกินที่น้อยที่สุด มีดังต่อไปนี้

1. เริ่มจากการหาค่าพารามิเตอร์ของตัวควบคุมกระแสก่อน โดยกำหนดค่ากระแสอาร์เมเจอร์อ้างอิงคงที่ ($I_{a_ref} = 13$) และกำหนดค่าพารามิเตอร์ของตัวควบคุมแบบพีไอ คือ $K_p = 0.129$, $K_i = 104.16$ (เป็นค่าที่หาได้จากวิธีที่เหมาะสมที่สุดแบบสมมาตร) ให้กับโปรแกรมจำลองผลการควบคุมมอเตอร์โดยตัวควบคุมแบบพีไอ (อยู่ในภาคผนวก ก)

2. สั่งการให้โปรแกรมทำงาน (run program) แล้วดูผลช่วงเวลารั้งขึ้น (t_r) คือการตอบสนองของความเร็วที่เริ่มจากความเร็วเป็นศูนย์จนกระทั่งความเร็วถึง 100 เรเดียน/วินาที ให้ได้ค่าช่วงเวลารั้งขึ้นที่น้อยที่สุด โดยการเปลี่ยนค่า K_p และ K_i ที่มากขึ้นและน้อยลง ซึ่งถ้าต้องการเปลี่ยนค่า K_p จะต้องกำหนดให้ค่า K_i คงที่ และในทางกลับกันถ้าต้องการเปลี่ยนค่า K_i จะต้องกำหนดให้ค่า K_p คงที่ และค่า K_p , K_i ที่เหมาะสมที่สุดจะต้องไม่ทำให้มอเตอร์มีกระแสอาร์เมเจอร์ที่สูงเกินกว่าค่าสูงสุดของมอเตอร์ทนได้

3. เมื่อหาค่าพารามิเตอร์ของตัวควบคุมกระแสจากขั้นตอนที่ 2 เรียบร้อยแล้ว (กำหนดค่าพารามิเตอร์นี้ลงในโปรแกรมจำลองผลการทำงาน) ในขั้นตอนนี้จะหาค่าพารามิเตอร์ของตัวควบคุมความเร็ว ซึ่งตามรูปที่ 3.12 เอาต์พุตของตัวควบคุมความเร็วจะเป็นค่ากระแสอ้างอิง (I_{a_ref}) ให้กับตัวควบคุมกระแส ดังนั้นจะต้องกำหนดให้ค่า I_{a_ref} = เอาต์พุตที่ได้จากตัวควบคุมความเร็ว และค่า K_p , K_i ของตัวควบคุมความเร็ว กำหนดเริ่มต้นที่ค่า $K_p = 115.37$ $K_i = 19.841$ (เป็นค่าที่นำมาจากการหาด้วยวิธีที่เหมาะสมที่สุดแบบสมมาตร) จากนั้นให้กำหนดค่า K_p หรือ K_i ให้คงที่หนึ่งค่าแล้วทำการเปลี่ยนแปลงค่า K_p หรือ K_i ให้มากขึ้นและน้อยลง แล้วดูผลการตอบสนองทางด้านความเร็วให้ได้ค่าช่วงเวลารั้งขึ้น ค่าเปอร์เซ็นต์การฟุ้งเกินและค่าช่วงเวลาเข้าที่ (t_s) น้อยที่สุด

ภาคผนวก ค. : โปรแกรมจำลองผลการควบคุมมอเตอร์โดยตัวควบคุมแบบพีไอ

โปรแกรมจำลองผลการควบคุมมอเตอร์ด้วยตัวควบคุมแบบพีไอ

```

PROGRAM PI-
Controller_For_Chopper_fed_Separately_Excited_DC_Motor_Drives;
(*---by Komkrit Prasertwong, 20 March 1997---*)
USES CRT; {use screen}
CONST
{-----Constant of Chopper-----}
f=200; { frequency = 200 Hz}
Time_Period1=1/f; { time period:T defin constant or f=200 Hz }
Time_End=5; { total time computation }
VAR
T,H:REAL; {T=Time, H=Step size of calulation}
J1,J2,J3,J4,K1,K2,K3,K4:REAL; {Variable of four-order Runge-
Kutta method}
L1,L2,L3,L4,M1,M2,M3,M4:REAL; {Variable of four-order Runge-
Kutta method}
X1,X2,X3,TL:REAL; { X1=Armature current, X2=Angular
velocity, X3=Angular displacement TL=Load torque }
Rf,Lf,Ra,La,kb,kt,J,B,V,Vt,Td:REAL; {Parameter of motor}
Sum_Ia,Sum_Td,Sum_Sp,Ia,Num_X,Sp:REAL;
{-----Variable of Chopper-----*}
Time_Period,Ton,Ton_New:REAL;
Time_Period_Old,Time_Period_New:REAL;

Duty_Cycle_Vt,Pulse_Width_Ton,Duty_Cycle,Pulse_Width_Ton_New
:REAL;
Duty_Cycle_Vf:REAL;
X2_OLD,X3_OLD:REAL;
Counter_V:INTEGER;
(*-----Variable of print to file-----*)

FILVAR1,FILVAR2,FILVAR3,FILVAR4,FILVAR5,FILVAR6,FILV
AR7,FILVAR8 :TEXT;
FILENAME1,FILENAME2,FILENAME3:STRING[14];

FILENAME4,FILENAME5,FILENAME6,FILENAME7,FILENAME8
:STRING[14];
T_Print:REAL;
(*-----end variable of print to file-----*)
(*-----Variable of PI-controller-----*)

Sp_ref,Ia_ref,kp_Sp,kp_Ia,ki_Sp,ki_Ia,E_Sp,E_Ia,V_Sp,V_Ia,ki:REAL;
I:INTEGER;
Vo_Ia,U2_Ia,U3_Ia,T_on,Ia_Tran,kd:REAL;
Vo_Sp,U2_Sp,U3_Sp,Sp_Tran,Vt_Max,V_Ia_Max:REAL;
(*-----End variable of PI-controller-----*)

(*----- FIRST STATE -----*)
FUNCTION FUNCX1D(T,X1,X2,X3,Ra,La,kb,V:REAL):REAL;
BEGIN
FUNCX1D:=(-Ra/La)*X1-(kb/La)*X2+(1/La)*V;
END;

(*----- SECOND STATE -----*)
FUNCTION FUNCX2D(T,X1,X2,X3,kt,J,B,TL:REAL):REAL;
BEGIN
FUNCX2D:=(kt/J)*X1-(B/J)*X2-(1/J)*TL;
END;

(*----- THIRD STATE -----*)
FUNCTION FUNCX3D(T,X1,X2,X3:REAL):REAL;
BEGIN
FUNCX3D:=X2;
END;
BEGIN {main program}
CLRSCR;
FILENAME1:='m1.DAT'; {data file}
WRITELN('WRITING DATA FILE : ',FILENAME1);
ASSIGN(FILVAR1,FILENAME1); { assign filename}
REWRITE(FILVAR1); {construct and open file}
(*-----*)

Ia:=0;Sp:=0;E_Ia:=0;E_Sp:=0;Ia_Tran:=0;Sp_Tran:=0;
U2_Ia:=0;U2_Sp:=0;
U3_Ia:=0;U3_Sp:=0;Vo_Ia:=0;V_Ia:=0;V_Sp:=0;
Ra:=1.0; {Ra=armature resistor(ohm)}
La:=0.046; {La=armature inductor(henry)}
kb:=0.55; {kb=back emf constant(V.sec/rad)}
kt:=0.55; {kt=torque constant(N.m./A)}
J:=0.093; {J=moment of inertia(kg-m^2)}
B:=0.008; {B=viscous friction(N.m.sec/rad)}
TL:=0.0; {TL=load torque(N-m)}
(*-----*)
T:=0.0; X1:=0; X2:=0; X3:=0;
H:=Time_Period1/50;
Sp_ref:=100*0.057;
kp_Sp:=115.37;ki_Sp:=19.841; kp_Ia:=0.129;ki_Ia:=104.16;
Time_Period:=Time_Period1;
Duty_Cycle:=0; Ton:=Duty_Cycle*Time_Period; {initail Ton/T
= 0.02 }
Sum_Ia:=0; Sum_Td:=0;Sum_Sp:=0;
T_on:=Ton; T_Print:=Time_Period1;
REPEAT {-----begin loop-----}
(*----- Command set of chopper -----*)

```

```

(*----- Armature voltage controller -----*)
IF (T<=Ton) OR ((T-Ton) < (H/10)) THEN {----duty interval----}
BEGIN
    IF T>Time_Period1 THEN V:=110.;
END;

IF (Ton<T) AND (T<Time_Period) AND ((T-Ton)>(H/10))
THEN
    {----freewheeling interval----}
BEGIN
    V:=0.0;
END;

IF (T>=Time_Period) OR (ABS(T-Time_Period)<(H/10)) THEN
BEGIN
    Ia:=Sum_Ia/Num_X; {Ia=average current}
    Td:=Sum_Td/Num_X; {Td=average developed torque}
    Sp:=Sum_Sp/Num_X; {Sp=average speed motor}
    Sum_Ia:=0;Sum_Td:=0;Sum_Sp:=0;
    Time_Period_Old:=Time_Period;
    Time_Period_New:=Time_Period_Old+Time_Period1;
    Time_Period:=Time_Period_New;
    (*-----Speed Controller-----*) (*-----PI Controller (kp(s
+ ki)/s)-----*)
    Sp_Tran:=0.057*Sp*(1-exp(-T/0.1)); {Sp_Tran is Sp pass
Tachogenerator kt=0.057 V/A
and tachogenerator filter
time constance =0.1}
    E_Sp:=Sp_ref-Sp_Tran;
    IF E_Sp>13 THEN E_Sp:=13; { E_Ia must varies +-13}
    IF E_Sp<-13 THEN E_Sp:=-13;
    U2_Sp:=kp_Sp*E_Sp;
    IF U2_Sp>13 THEN U2_Sp:=13; {U2_Sp varies +-13 V}
    IF U2_Sp<-13 THEN U2_Sp:=-13;
    U3_Sp:=kp_Sp*ki_Sp*T_on*E_Sp+Vo_Sp;
    IF U3_Sp>13 THEN U3_Sp:=13; { U3_Sp varies +-13 V}
    IF U3_Sp<-13 THEN U3_Sp:=-13;
    V_Sp:=U2_Sp+U3_Sp;
    IF V_Sp>13 THEN V_Sp:=13; {V_Sp varies 0 to +13 V}
    IF V_Sp<0 THEN V_Sp:=0;
    Ia_ref:=V_Sp;
    IF Ia_ref>13 THEN Ia_ref:=13; {Current Limit Reference} {for
a current limit of 26 A}

    (*-----End PI Speed Controller -----*)
    (*-----Current Controller-----*) (*-----PI Controller (kp(s + ki)/s)
-----*)
    Ia_Tran:=0.5*Ia; {Ia_Tran is Ia pass current transducer kr=0.5
V/A}

    E_Ia:=Ia_ref-Ia_Tran; { E_Ia must varies +-13}
    IF E_Ia>13 THEN E_Ia:=13;
    IF E_Ia<-13 THEN E_Ia:=-13;
    U2_Ia:=kp_Ia*E_Ia;
    IF U2_Ia>13 THEN U2_Ia:=13;
    IF U2_Ia<-13 THEN U2_Ia:=-13;
    U3_Ia:=kp_Ia*ki_Ia*T_on*E_Ia+Vo_Ia;
    IF U3_Ia>13 THEN U3_Ia:=13;
    IF U3_Ia<-13 THEN U3_Ia:=-13;
    V_Ia:=U2_Ia+U3_Ia;
    GOTOXY(1,4);WRITE(' Ia =',Ia:10:4,' sec. '); GOTOXY(1,5);
WRITE(' Sp =',Sp:10:4,' sec. ');
    GOTOXY(1,6);WRITE(' Sp_ref =',Sp_ref:10:4,' sec. ');
    GOTOXY(1,7);WRITE(' Sp_Tran =',Sp_Tran:10:4,' sec. ');
    GOTOXY(1,8);WRITE(' E_Sp =',E_Sp:10:4,' sec. ');
    GOTOXY(1,9);WRITE(' T_on =',T_on:10:4,' sec. ');
    GOTOXY(1,10);WRITE(' U2_Sp =',U2_Sp:10:4,' sec. ');
    GOTOXY(1,11);WRITE(' U3_Sp =',U3_Sp:10:4,' sec. ');
    GOTOXY(1,12);WRITE(' Vo_Sp =',Vo_Sp:10:4,' sec. ');
    GOTOXY(1,13);write(' V_Sp =',V_Sp:10:4);
    GOTOXY(48,6);WRITE(' Ia_ref =',Ia_ref:10:4,' sec. ');
    GOTOXY(48,7);WRITE(' Ia_Tran =',Ia_Tran:10:4,' sec. ');
    GOTOXY(48,8);WRITE(' E_Ia =',E_Ia:10:4,' sec. ');
    GOTOXY(48,9);WRITE(' T_on =',T_on:10:4,' sec. ');
    GOTOXY(48,10);WRITE(' U2_Ia =',U2_Ia:10:4,' sec. ');
    GOTOXY(48,11);WRITE(' U3_Ia =',U3_Ia:10:4,' sec. ');
    GOTOXY(48,12);WRITE(' Vo_Ia =',Vo_Ia:10:4,' sec. ');
    GOTOXY(48,13);write(' V_Ia =',V_Ia:10:4);
    Vt_Max:=0.99*110; {Maximum armature vltage
Vt=0.99*110=108.9}
    V_Ia_Max:=13; {Maximum control voltage :V_Ia or Ec}
    kd:=Vt_Max/V_Ia_Max; {kd=linear relationship between control
voltage V_Ia;(Ec)
and armature voltage Vt=Duty_Cycle*V;(Ea) }
    IF V_Ia>=13 THEN V_Ia:=13;
    IF V_Ia<=0 THEN V_Ia:=0;
    Duty_Cycle:=(kd*V_Ia)/110; {convert Vt to Duty_Cycle}
    GOTOXY(48,14);writeln(' Duty_Cycle =',Duty_Cycle:7:4);
    (*-----End Current Controller-----*)
    Pulse_Width_Ton_New:=Duty_Cycle*Time_Period1;
    Pulse_Width_Ton:=Pulse_Width_Ton_New;
    Ton_New:=Time_Period_Old+Pulse_Width_Ton;
    Ton:=Ton_New;
    IF T>Time_Period1 THEN V:=110;
    Counter_V:=0;
    Vo_Ia:=U3_Ia;
    Vo_Sp:=U3_Sp;
    T_on:=Pulse_Width_Ton_New;

```

```

(*----- Over current protection-----*)
IF Ia_Tran > 0.5*40 THEN {Transient operation motor current can
                           be allowed 2-3.5 times the rated current}
    BEGIN
        Ton:=Time_Period_Old;
        WRITELN('Over current Ia = ',Ia:8:4);
        Sound(220);    { Beep }
        Delay(200);    { For 200 ms }
        NoSound;    { Relief! }
        {   Halt(1); { Halt right here! }
        END;
        (*-----end-----*)
    END;
    IF T>3.0 THEN TL:=9.0; { Step Load torque (TL) is applied at
    T=3.0 sec}

(*----- Over current protection-----*)
IF Ia_Tran > 0.5*40 THEN
    BEGIN {Ia_Maximum be 2 times the rated current }
        V:=0;
        WRITELN('Over current Ia = ',Ia:8:4);
        Sound(220);    { Beep }    Delay(200);    { For 200 ms }
        NoSound;    { Relief! }
        END;
        (*-----Set command calculation Fourth-order runge-kutta method---
        *)
        K1:=H*FUNCX1D(T,X1,X2,X3,Ra,La,kb,V);
        L1:=H*FUNCX2D(T,X1,X2,X3,kt,J,B,TL);
        M1:=H*FUNCX3D(T,X1,X2,X3);
        K2:=H*FUNCX1D(T+0.5*H,X1+0.5*K1,X2+0.5*
        L1,X3+0.5*M1,Ra,La,kb,V);
        L2:=H*FUNCX2D(T+0.5*H,X1+0.5*K1,X2+0.5*
        L1,X3+0.5*M1,kt,J,B,TL);
        M2:=H*FUNCX3D(T+0.5*H,X1+0.5*K1,X2+0.5*
        L1,X3+0.5*M1);
        K3:=H*FUNCX1D(T+0.5*H,X1+0.5*K2,X2+0.5*
        L2,X3+0.5*M2,Ra,La,kb,V);
        L3:=H*FUNCX2D(T+0.5*H,X1+0.5*K2,X2+0.5*
        L2,X3+0.5*M2,kt,J,B,TL);
        M3:=H*FUNCX3D(T+0.5*H,X1+0.5*K2,X2+0.5*
        L2,X3+0.5*M2);
        K4:=H*FUNCX1D(T+H,X1+K3,X2+L3,X3+M3,Ra,La,kb,V);
        L4:=H*FUNCX2D(T+H,X1+K3,X2+L3,X3+M3,kt,J,B,TL);
        M4:=H*FUNCX3D(T+H,X1+K3,X2+L3,X3+M3);
        X1:=X1+(K1+2*K2+2*K3+K4)/6;
        X2:=X2+(L1+2*L2+2*L3+L4)/6;
        X3:=X3+(M1+2*M2+2*M3+M4)/6;
        (*-----Comput Error Armature Current-----*)
        (*-----Compute Developed Torque-----*)

```

```

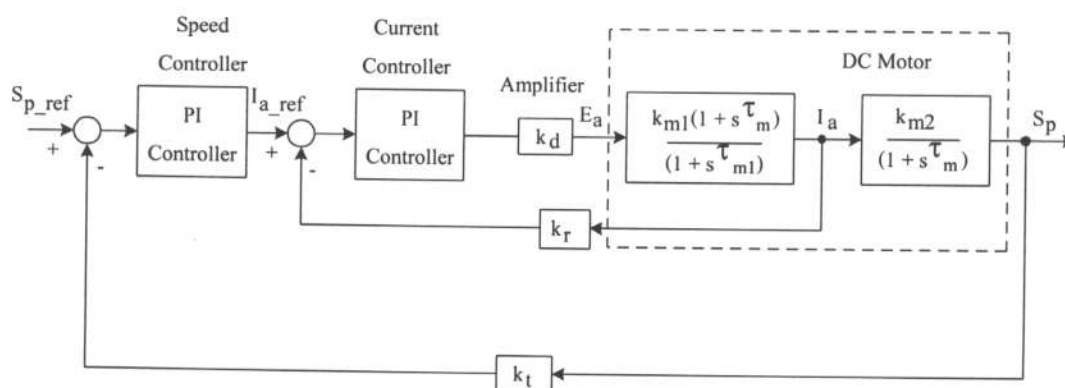
        Td:=kt*X1;    {Td=Develop torque}
        (*-----*)
        (*-----Compute Average Armature Current:Ia-----*)
        IF (T <=Time_Period) OR (ABS(T-Time_Period)<(H/10)) THEN
            BEGIN
                Sum_Ia:= X1+Sum_Ia;
                Sum_Td:= Td+Sum_Td;
                Sum_Sp:= X2+Sum_Sp;
                Num_X:=Time_Period/H;
            END;
            (*-----end-----*)
            (*-----compute average armature voltage-----*)
            IF V=110 THEN
                BEGIN
                    Counter_V:=Counter_V+1;
                END ELSE
                BEGIN
                    Duty_Cycle_Vt:=((Counter_V-1)*H)/Time_Period1;
                    IF T>Time_Period1 THEN
                        BEGIN
                            Vt:=Duty_Cycle_Vt*110;
                            IF Vt<0 THEN Vt:=0; {Vt=Terminal voltage or average
                            voltage}
                        END
                    END;
                    (*-----END-----*)
                    IF (T>=0.0) AND (T<5.0) THEN {Given print follow condition
                    IF statement }
                        BEGIN
                            IF T>=T_Print THEN BEGIN
                                WRITELN(FILVAR1,T:11:5,Vt:10:4,Duty_Cycle:10:4,
                                Ia:11:4,Sp:11:4);
                                T_Print:=T_Print+Time_Period1; END;
                            END;
                                GOTOXY(28,15);WRITE(' T = ',T:11:5,' sec. ');
                                GOTOXY(28,16);WRITE('Duty Cycle = ',Duty_Cycle:11:5,' sec. ');
                                GOTOXY(28,17);WRITE('Armature voltage:Vt= ',Vt:11:5,' V. ');
                                GOTOXY(28,18);WRITE('Current Transducer= ',Ia_Tran:11:5,' A.
                                ');
                                GOTOXY(28,19);WRITE('Average Current= ',Ia:11:5,' A. ');
                                GOTOXY(28,20);WRITE('Actual Current= ',X1:11:5,' A. ');
                                GOTOXY(28,21);WRITE('Angular velocity = ',X2:11:5,' rad/sec');
                                GOTOXY(28,22);WRITELN('Angular displacement = ',X3:11:5,'
                                rad');
                                T:=T+H;
                                UNTIL T>=Time_End; {add terminal voltage:Vt until Vt>=100
                                volt}
                                {-----}
                                CLOSE(FILVAR1); END.

```

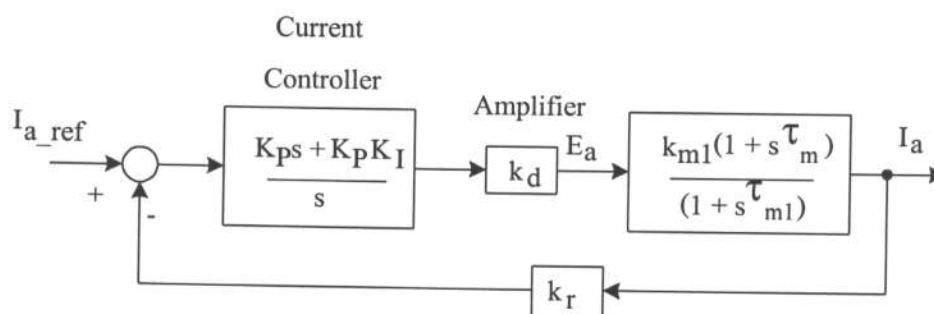
ภาคผนวก ง. : การหาตำแหน่งโพลวงปิดของระบบขับเคลื่อน

การหาตำแหน่งโพลของระบบขับเคลื่อน

จากแผนภาพการควบคุมระบบขับเคลื่อนโดยตัวควบคุมแบบพีไอ ที่ได้แสดงไว้ในรูปที่ 3.12 สามารถเขียนเป็นแผนภาพระบบการขับเคลื่อนมอเตอร์ที่อยู่ในรูปของฟังก์ชันถ่ายโอน ดังแสดงในรูปที่ ง.1 ซึ่งบล็อกไดอะแกรมที่อยู่ภายในเส้นประ แทนแบบจำลองทางคณิตศาสตร์ของมอเตอร์แบบกระตุ้นฟลักซ์แยกส่วน ตัวขยาย (k_d) นั้นหมายถึงวงจรสับไฟฟ้าซึ่งทำหน้าที่ขยายสัญญาณเอาต์พุตที่ได้จากตัวควบคุมกระแส และสัญญาณเอาต์พุตของตัวขยายจะเป็นแรงดัน E_a ที่ป้อนให้กับมอเตอร์ จากรูปที่ ง.1 เริ่มชวยวงรอบการควบคุมกระแสก่อนดังแสดงในรูปที่ ง.2

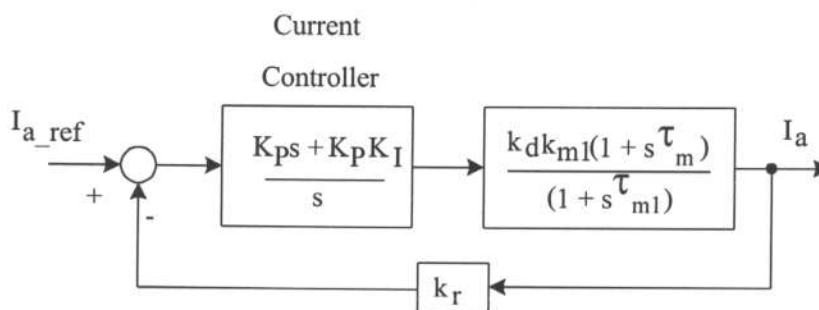


รูปที่ ง.1 แผนภาพระบบการควบคุมความเร็วของมอเตอร์



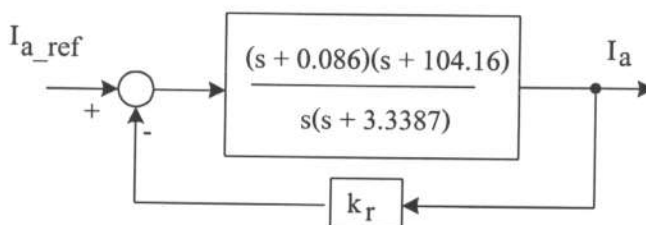
รูปที่ ง.2 แผนภาพระบบขับเคลื่อนมอเตอร์ในส่วนการควบคุมกระแสวงรอบปิด

ในรูปที่ ง.2 ขั้วบล็อกของตัวขยาย (k_d) รวมเข้ากับบล็อกความสัมพันธ์ระหว่าง E_a กับ I_a และเขียนรูปใหม่ แสดงดังรูปที่ ง.3 (สำหรับโปรแกรมที่ใช้ขั้วบล็อกต่างๆในรูปที่ ง.1 เพื่อหาความสัมพันธ์ระหว่าง S_{p_ref} กับ S_p แบบวงรอบเปิดได้แสดงไว้ในส่วนท้ายของภาคผนวกนี้)

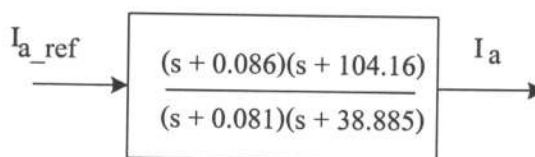


รูปที่ ๓.3 แผนภาพระบบขับเคลื่อนมอเตอร์ในส่วนการควบคุมกระแสวงปิด

จากรูปที่ ๓.3 ทำการรวมบล็อกตัวควบคุมกระแสกับบล็อกความสัมพันธ์ระหว่างเอาต์พุตของตัวควบคุมกระแส กับ I_a (ค่าพารามิเตอร์ของตัวควบคุมกระแสและตัวควบคุมความเร็ว ใช้ค่าที่หาได้จากวิธีเหมาะสมที่สุดแบบสมมาตร และค่าพารามิเตอร์ต่างๆของระบบขับเคลื่อนนำมาจากภาคผนวก ข.) ผลลัพธ์จากการรวมบล็อก แสดงดังรูปที่ ๓.4 และเมื่อยุบบล็อกต่อไปอีกเพื่อหาความสัมพันธ์ระหว่าง I_{a_ref} กับ I_a ผลลัพธ์ที่ได้แสดงดังรูปที่ ๓.5



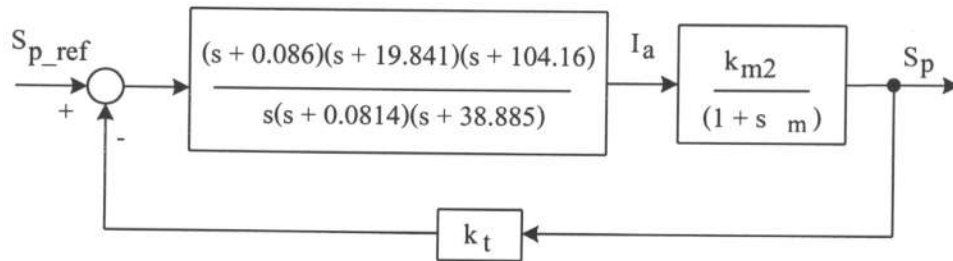
รูปที่ ๓.4 แผนภาพระบบขับเคลื่อนมอเตอร์ในส่วนการควบคุมกระแสวงปิด



รูปที่ ๓.5 แผนภาพบล็อกความสัมพันธ์ระหว่าง I_{a_ref} กับ I_a

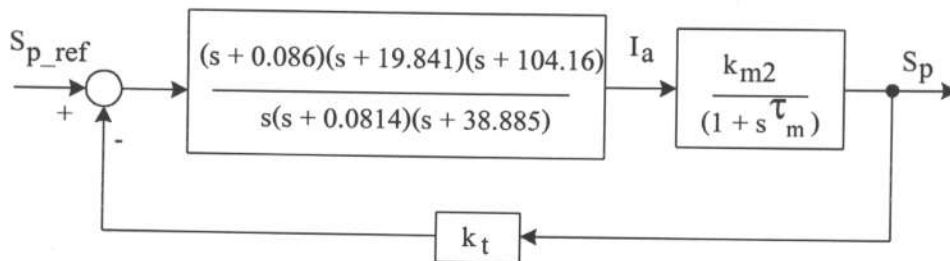
จากรูปที่ ๓.5 ความสัมพันธ์ของกระแสอาร์เมเจอร์อ้างอิงกับกระแสอาร์เมเจอร์ มีจำนวนโพล คือ $s = -0.081, -38.885$ และศูนย์ คือ $s = -0.086, -104.16$

เมื่อได้ความสัมพันธ์ระหว่าง I_{a_ref} กับ I_a ในรูปที่ ง.5 แทนฟังก์ชันถ่ายโอนลงในรูปที่ ง.1 และเขียนแผนภาพบล็อกการควบคุมใหม่แสดงดังรูปที่ ง.6



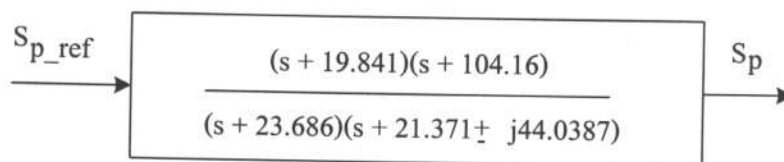
รูปที่ ง.6 แผนภาพบล็อกการควบคุมความเร็วมอเตอร์เมื่อยุบการควบคุมกระแสปิด

จากรูปที่ ง.6 รวมบล็อกตัวควบคุมความเร็วเข้ากับบล็อกความสัมพันธ์ระหว่าง I_{a_ref} กับ I_a และเขียนแผนภาพการควบคุมใหม่ ดังแสดงในรูปที่ ง.7



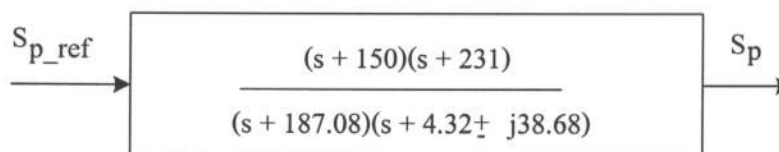
รูปที่ ง.7 แผนภาพบล็อกการควบคุมความเร็วมอเตอร์

จากรูปที่ ง.7 ยุบการควบคุมความเร็ววงรอบปิด ได้บล็อกฟังก์ชันถ่ายโอน ดังแสดงในรูปที่ ง.8 ซึ่งตำแหน่งของโพล คือ $s = -23.686, -21.371 \pm j44.0387$ และศูนย์ คือ $s = -19.841, -104.16$



รูปที่ ง.8 แผนภาพบล็อกความสัมพันธ์ระหว่าง S_{p_ref} กับ S_p (PI1)

จากรูปที่ ๙.๘ เป็นแผนภาพบล็อกความสัมพันธ์ระหว่าง S_{p_ref} กับ S_p ซึ่งค่าพารามิเตอร์ของตัวควบคุมกระแสและตัวควบคุมความเร็ว (ใช้ตัวควบคุมแบบพีไอ) หาโดยวิธีเหมาะสมที่สุดแบบสมมาตร (PI1) และเมื่อใช้ค่าพารามิเตอร์ของตัวควบคุมกระแสและตัวควบคุมความเร็ว ที่หาโดยวิธีปรับเปลี่ยนค่าและดูการตอบสนองของระบบที่ดีที่สุด (PI2) จะได้แผนภาพบล็อกความสัมพันธ์ระหว่าง S_{p_ref} กับ S_p ดังแสดงในรูปที่ ๙.๙ ซึ่งตำแหน่งโพล คือ $s = -187.08, -4.32 \pm j38.68$ และศูนย์ คือ $s = -150, -231$



รูปที่ ๙.๙ แผนภาพบล็อกความสัมพันธ์ระหว่าง S_{p_ref} กับ S_p (PI2)

โปรแกรม MATLAB ที่ใช้ในการหาความสัมพันธ์ ระหว่าง S_{p_ref} กับ S_p แบบวงรอบเปิด

```

echo on
% DC separately excited motor
% 1.865kW rated power,
%rated speed 188.49 rad/sec
%rated armature voltage 110V
%and rated current 20 A.
Ra=1.0; La=0.046; J=0.093; B=0.008; kb=0.55;
ki=0.55;
ta=La/Ra
tm=J/B
tm1=Ra*B*tm/(Ra*B+ki^2)
km1=B/(Ra*B+ki^2)
km2=ki/B
km=km1*km2
kd=110/13;
cl=kd*km1;

KP=1.0; KI=231;
%KP=0.129; KI=104.16;

num1=[KP KP*KI]
den1=[1 0]

% Current Transducer (kr)
numh=[0.5]; denh=[1];

% Armature circuit
num2=c1*[tm 1]
den2=[tm1 1]

% Step 1
[num3,den3]=series(num1,den1,num2,den2)

% Step 2
[num4,den4]=feedback(num3,den3,numh,denh,-1)

```

```

[p,z]=pzmap(num4,den4)                                % end program
pause

% Speed Controller (PI-Controller)
KP_Sp=15; KI_Sp=150;
%KP_Sp=115.37; KI_Sp=19.841;
num5=[KP_Sp  KP_Sp*KI_Sp]
den5=[1  0];

% Step 3
[num6,den6]=series(num5,den5,num4,den4);

% Mechanical circuit
num7=[0  km2]
den7=[tm  1]

% Step 4
[num8,den8]=series(num6,den6,num7,den7);
printsys(num8,den8)
[p,z]=pzmap(num8,den8)
pause

% Speed Transducer ( $k_t$ )
num9=[0.057]; den9 =[1];

% Step 5
[num10,den10]=feedback(num8,
den8,num9,den9,-1)
printsys(num10,den10);
[p,z]=pzmap(num10,den10)
step(num10,den10)
pause
[num,den]=minreal(num10,den10);
[p,z]=pzmap(num,den)

```

ภาคผนวก จ. : โปรแกรมจำลองผลการควบคุมมอเตอร์โดยตัวควบคุมแบบฟัซซี่ลอจิก

โปรแกรมจำลองผลการควบคุมมอเตอร์โดยมีตัวควบคุมแบบฟัซซี่ลอจิก

```

PROGRAM
Fuzzy_Logic_Controller_for_Chopper-fed_Separately_Excited_DC_M
otor_Drives;
(*----by Komkrit Prasertwong, 25 March 1997----*)
USES CRT; {use screen}
CONST
{-----Constant of Chopper-----}
    { frequency = 200 Hz }
    {Ton1=0.0001;    { Initial Ton=Ton1 }
    f=200;    Time_Period1=1/f; { time period:T defin constant or
f=1428.57 Hz }
    Time_End=5 ;    { total time computation }
    {-----}
VAR  Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11:REAL;
    T,H:REAL;    {T=Time, H=Step size of calulation}
    J1,J2,J3,J4,K1,K2,K3,K4,L1,L2,L3,L4,M1,M2,M3,M4:REAL;
{variable of fourth-order method}
    X1,X2,X3,X4,TL:REAL; { X1=field current, X2=armature current,
X3=angular velocity, X4=angular displacement, TL=load torque }
    Rf,Lf,Ra,La,kb,kt,J,B,V,Vt,Td,Vf,Vfield:REAL; {parameter of
motor}

Sum_Ia,Sum_Td,Sum_Sp,Ia_Tran,Num_X,Sp_Tran,Ia,Sp,kp,Ti,Ia_Ref,
E_Ia,E_Sp,Sp_ref:REAL;
{*-----Variable of Chopper-----*}
    Time_Period,Ton,Ton_New,
Time_Period_Old,Time_Period_New:REAL;

Duty_Cycle_Vt,Pulse_Width_Ton,Duty,Pulse_Width_Ton_New:REAL
;
    Counter_V:INTEGER;
(*-----Variable of Print to file-----*)

FILVAR1,FILVAR2,FILVAR3,FILVAR4,FILVAR5,FILVAR6,FILV
AR7:TEXT;
    FILENAME1,FILENAME2,FILENAME3:STRING[14];

FILENAME4,FILENAME5,FILENAME6,FILENAME7,FILENAME8
:STRING[14];
    T_Print:REAL;
*-----variable of Fuzzy-----*)
{---variable of Fuzzifier-----}
    MEW_NB_Sp,MEW_NM_Sp,MEW_NS_Sp,
MEW_ZO_Sp,MEW_PS_Sp,MEW_PM_Sp,MEW_PB_Sp,
MEW_NB_Ia,MEW_NM_Ia,MEW_NS_Ia,MEW_ZO_Ia,
MEW_PS_Ia,MEW_PM_Ia,MEW_PB_Ia:REAL;

{-----end variable of Fuzzifier-----}
{---variable of Defuzzifier-----}
    Sum_W,Sum_Mew:REAL;
{-----end variable of Defuzzifier-----}
(*-----end variable of Fuzzy-----*)
(*----- FIRST STATE -----*)
FUNCTION FUNCX1D(T,X1,X2,X3,Ra,La,kb,V:REAL):REAL;
BEGIN
    FUNCX1D:=(-Ra/La)*X1-(kb/La)*X2+(1/La)*V;
END;
(*----- SECOND STATE -----*)
FUNCTION FUNCX2D(T,X1,X2,X3,kt,J,B,TL:REAL):REAL;
BEGIN
    FUNCX2D:=(kt/J)*X1-(B/J)*X2-(1/J)*TL;
END;
(*----- THIRD STATE -----*)
FUNCTION FUNCX3D(T,X1,X2,X3:REAL):REAL;
BEGIN
    FUNCX3D:=X2;
END;

PROCEDURE MEMBERSHIP_Ia(E_Ia:REAL;VAR
NB,NM,NS,ZO,PS,PM,PB:REAL);
VAR
    MEW_NB_Ia,MEW_NM_Ia,MEW_NS_Ia,MEW_ZO_Ia,
MEW_PS_Ia,MEW_PM_Ia,MEW_PB_Ia:REAL;
    B,C,D,E,F,G,H,I:REAL;
BEGIN {--BEGIN PROCEDURE--}
    B:=1.5;C:=1.0;D:=0.5;E:=0;F:=0.5;G:=1;H:=1.5;

    MEW_NB_Ia:=0;MEW_NM_Ia:=0;MEW_NS_Ia:=0;MEW_ZO_Ia:=0;
    MEW_PS_Ia:=0;MEW_PM_Ia:=0;MEW_PB_Ia:=0;
    {---FUNCTION NB OF SPEED ERROR---}
    IF (E_Ia>=B) AND (E_Ia<=C) THEN MEW_NB_Ia:=(C-E_Ia)/(C-
B);
    IF (E_Ia<B) THEN MEW_NB_Ia:=1;
    {---END NB ---}
    GOTOXY(45,6);WRITELN('MEW_NB_Ia = ',MEW_NB_Ia:5);

    {--- FUNCTION NM OF SPEED ERROR ---}
    IF (E_Ia>=B) AND (E_Ia<=C) THEN MEW_NM_Ia:=(E_Ia-B)/
(C-B);
    IF (E_Ia>=C) AND (E_Ia<=D) THEN MEW_NM_Ia:=(D-E_Ia)/
(D-C);
    GOTOXY(45,7);WRITELN('MEW_NM_Ia = ',MEW_NM_Ia:5:5);
    {--- END NM ---}

```

```

{---- FUNCTION NS OF SPEED ERROR ----}
IF (E_Ia>=C) AND (E_Ia<=D) THEN MEW_NS_Ia:=(E_Ia-C)/(D-
C);
IF (E_Ia>=D) AND (E_Ia<=E) THEN MEW_NS_Ia:=(E-E_Ia)/(E-
D);
GOTOXY(45,8);WRITELN('MEW_NS_Ia = ',MEW_NS_Ia:5:5);
{---- END NS ----}

{---- FUNCTION ZO OF SPEED ERROR ----}
IF (E_Ia>=D) AND (E_Ia<=E) THEN MEW_ZO_Ia:=(E_Ia-D)/(E-
D);
IF (E_Ia>=E) AND (E_Ia<=F) THEN MEW_ZO_Ia:=(F-E_Ia)/(F-
E);
GOTOXY(45,9);WRITELN('MEW_ZO_Ia = ',MEW_ZO_Ia:5:5);
{---- END ZO ----}

{---- FUNCTION PS OF SPEED ERROR ----}
IF (E_Ia>=E) AND (E_Ia<=F) THEN MEW_PS_Ia:=(E_Ia-E)/(F-
E);
IF (E_Ia>=F) AND (E_Ia<=G) THEN MEW_PS_Ia:=(G-E_Ia)/(G-
F);
GOTOXY(45,10);WRITELN('MEW_PS_Ia = ',MEW_PS_Ia:5:5);
{---- END PS ----}

{---- FUNCTION PM OF SPEED ERROR ----}
IF (E_Ia>=F) AND (E_Ia<=G) THEN MEW_PM_Ia:=(E_Ia-F)/(G-
F);
IF (E_Ia>=G) AND (E_Ia<=H) THEN MEW_PM_Ia:=(H-E_Ia)/(
H-G);
GOTOXY(45,11);WRITELN('MEW_PM_Ia = ',MEW_PM_Ia:5:5);
{---- END PM ----}

{---- FUNCTION PB OF SPEED ERROR ----}
IF (E_Ia>=G) AND (E_Ia<=H) THEN MEW_PB_Ia:=(E_Ia-G)/(H-
G);
IF E_Ia>H THEN MEW_PB_Ia:=1;
GOTOXY(45,12);WRITELN('MEW_PB_Ia = ',MEW_PB_Ia:5:5);
{---- END PB ----}
NB:=MEW_NB_Ia; NM:=MEW_NM_Ia; NS:=MEW_NS_Ia;
ZO:=MEW_ZO_Ia; PS:=MEW_PS_Ia; PM:=MEW_PM_Ia;
PB:=MEW_PB_Ia;
END; {--PROCEDURE--}

PROCEDURE MEMBERSHIP_Sp(E_Sp:REAL;VAR
NS,ZO,PS,PM,PB:REAL);
VAR
MEW_NS_Sp,MEW_ZO_Sp,MEW_PS_Sp,
MEW_PM_Sp,MEW_PB_Sp:REAL;

```

```

D,E,F,G,H,I:REAL;
BEGIN {--BEGIN PROCEDURE--}
D:=2;E:=0;F:=2;G:=4;H:=6;

MEW_NS_Sp:=0;MEW_ZO_Sp:=0;MEW_PS_Sp:=0;MEW_PM_Sp:=
0;MEW_PB_Sp:=0;

{---- FUNCTION NS OF SPEED ERROR ----}
IF (E_Sp<=D) THEN MEW_NS_Sp:=1;
IF (E_Sp>=D) AND (E_Sp<=E) THEN MEW_NS_Sp:=(E-E_Sp)/
(E-D);
GOTOXY(1,6);WRITELN('MEW_NS_Sp = ',MEW_NS_Sp:5:5);
{---- END NS ----}

{---- FUNCTION ZO OF SPEED ERROR ----}
IF (E_Sp>=D) AND (E_Sp<=E) THEN MEW_ZO_Sp:=(E_Sp-D)/
(E-D);
IF (E_Sp>=E) AND (E_Sp<=F) THEN MEW_ZO_Sp:=(F-E_Sp)/
(F-E);
GOTOXY(1,7);WRITELN('MEW_ZO_Sp = ',MEW_ZO_Sp:5:5);
{---- END ZO ----}

{---- FUNCTION PS OF SPEED ERROR ----}
IF (E_Sp>=E) AND (E_Sp<=F) THEN MEW_PS_Sp:=(E_Sp-E)/
(F-E);
IF (E_Sp>=F) AND (E_Sp<=G) THEN MEW_PS_Sp:=(G-E_Sp)/
(G-F);
GOTOXY(1,8);WRITELN('MEW_PS_Sp = ',MEW_PS_Sp:5:5);
{---- END PS ----}

{---- FUNCTION PM OF SPEED ERROR ----}
IF (E_Sp>=F) AND (E_Sp<=G) THEN MEW_PM_Sp:=(E_Sp-F)/
(G-F);
IF (E_Sp>=G) AND (E_Sp<=H) THEN MEW_PM_Sp:=(H-E_Sp)/
(H-G);
GOTOXY(1,9);WRITELN('MEW_PM_Sp = ',MEW_PM_Sp:5:5);
{---- END PM ----}

{---- FUNCTION PB OF SPEED ERROR ----}
IF (E_Sp>=G) AND (E_Sp<=H) THEN MEW_PB_Sp:=(E_Sp-G)/
(H-G);
IF E_Sp>H THEN MEW_PB_Sp:=1;
GOTOXY(1,10);WRITELN('MEW_PB_Sp = ',MEW_PB_Sp:5:5);
{---- END PB ----}
NS:=MEW_NS_Sp; ZO:=MEW_ZO_Sp; PS:=MEW_PS_Sp;
PM:=MEW_PM_Sp; PB:=MEW_PB_Sp;
END; {--PROCEDURE--}

PROCEDURE DEFUZZI_DUTY(MEW_VS_Duty,MEW_S_Duty,

```

```

MEW_QS_Duty,MEW_SM_Duty,MEW_M_Duty,MEW_MB_Duty,
MEW_QB_Duty, MEW_B_Duty,MEW_VB_Duty,
MEW_AL_Duty:REAL; VAR Duty:REAL);
VAR
(*-----Variable of Defuzzification-----*)
A,B,C,D,E,F,G,H,I,J,K,L, x1,x2,x3,x4,y1,y2,y3,y4,Ycoa,
m1,m2,b1,b2:REAL;

W_VS,W_S,W_QS,W_SM,W_M,W_MB,W_QB,W_B,W_VB,W_AL:
REAL;
(*-----End variable of defuzzification ---*)

BEGIN (--PROCEDURE--)
A:=0;B:=0.1;C:=0.2;D:=0.3;E:=0.4;F:=0.5;G:=0.6;H:=0.65;
I:=0.8;J:=0.9;K:=0.99;L:=1.0;
W_VS:=0;W_S:=0;W_QS:=0;W_SM:=0;W_M:=0;
W_MB:=0;W_QB:=0;W_B:=0;W_VB:=0;W_AL:=0;
(*-- Very Small Duty Cycle ----*)
IF Mew_VS_Duty>0 THEN BEGIN
x1:=A; x2:=C-Mew_VS_Duty*(C-B); x3:=C; x4:=A;
y1:=0; y3:=0; y2:=Mew_VS_Duty; y4:=Mew_VS_Duty; m1:=(y2-
y1)/(x2-x1); m2:=(y4-y3)/(x4-x3);
b1:=(y2*x1-y1*x2)/(x1-x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_VS:=Mew_VS_Duty*Duty;
END;
(*---- End Very Small Duty Cycle ----*)

(*----- Small Duty Cycle -----*)
IF Mew_S_Duty>0 THEN BEGIN
x1:=B; x2:=D-Mew_S_Duty*(D-C); x3:=D; x4:=Mew_S_Duty*(C-
B)+B;
y1:=0; y3:=0; y2:=Mew_S_Duty; y4:=Mew_S_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_S:=Mew_S_Duty*Duty;
END;
(*----- End Small Duty Cycle -----*)
(*----- Quite Small Duty Cycle -----*)
IF Mew_QS_Duty>0 THEN BEGIN
x1:=C; x2:=E-Mew_QS_Duty*(E-D); x3:=E; x4:=Mew_QS_Duty*
(D-C)+C;
y1:=0; y3:=0; y2:=Mew_QS_Duty; y4:=Mew_QS_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_QS:=Mew_QS_Duty*Duty;
END;
(*----- End Quite Small Duty Cycle -----*)

(*----- Small to Medium Duty Cycle -----*)
IF Mew_SM_Duty>0 THEN BEGIN
x1:=D; x2:=F-Mew_SM_Duty*(F-E); x3:=F; x4:=Mew_SM_Duty*
(E-D)+D;
y1:=0; y3:=0; y2:=Mew_SM_Duty; y4:=Mew_SM_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_SM:=Mew_SM_Duty*Duty;
END;
(*----- End Small to Medium Duty Cycle -----*)
(*----- Medium Duty Cycle -----*)
IF Mew_M_Duty>0 THEN BEGIN
x1:=E; x2:=G-Mew_M_Duty*(G-F); x3:=G; x4:=Mew_M_Duty*(F-
E)+E;
y1:=0; y3:=0; y2:=Mew_M_Duty; y4:=Mew_M_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_M:=Mew_M_Duty*Duty;
END;
(*----- End Medium Duty Cycle -----*)

(*----- Medium to Big Duty Cycle -----*)
IF Mew_MB_Duty>0 THEN BEGIN
x1:=F; x2:=H-Mew_MB_Duty*(H-G); x3:=H; x4:=Mew_MB_Duty*
(G-F)+F;
y1:=0; y3:=0; y2:=Mew_MB_Duty; y4:=Mew_MB_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_MB:=Mew_MB_Duty*Duty;
END;
(*-----End Medium to Big Duty Cycle -----*)

(*----- Quite Big Duty Cycle -----*)
IF Mew_QB_Duty>0 THEN BEGIN
x1:=G; x2:=I-Mew_QB_Duty*(I-H); x3:=I; x4:=Mew_QB_Duty*(H-
G)+G;
y1:=0; y3:=0; y2:=Mew_QB_Duty; y4:=Mew_QB_Duty;

```

```

m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_QB:=Mew_QB_Duty*Duty;
END;
(*----- End Quite Big Duty Cycle -----*)

(*----- Big Duty Cycle -----*)
IF Mew_B_Duty>0 THEN BEGIN
x1:=H; x2:=J-Mew_B_Duty*(J-I); x3:=J; x4:=Mew_B_Duty*(I-H)+
H;
y1:=0; y3:=0; y2:=Mew_B_Duty; y4:=Mew_B_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_B:=Mew_B_Duty*Duty;
END;
(*----- End Big Duty Cycle -----*)

(*----- Very Big Duty Cycle -----*)
IF Mew_VB_Duty>0 THEN BEGIN
x1:=I; x2:=K-Mew_VB_Duty*(K-J); x3:=K; x4:=Mew_VB_Duty*(J-
I)+I;
y1:=0; y3:=0; y2:=Mew_VB_Duty; y4:=Mew_VB_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_VB:=Mew_VB_Duty*Duty;
END;
(*----- End Very Big Duty Cycle -----*)
(*----- About Limit Duty Cycle -----*)
IF Mew_AL_Duty>0 THEN BEGIN
x1:=J; x2:=L; x3:=L; x4:=Mew_AL_Duty*(K-J)+J;
y1:=0; y3:=0; y2:=Mew_AL_Duty; y4:=Mew_AL_Duty;
m1:=(y2-y1)/(x2-x1); m2:=(y4-y3)/(x4-x3); b1:=(y2*x1-y1*x2)/(x1-
x2); b2:=(y4*x3-y3*x4)/(x3-x4);
Ycoa:=(b1*m2-m1*b2)/(m2-m1); Xcoa:=(b2-b1)/(m1-m2);
Duty:=Xcoa;
W_AL:=Mew_AL_Duty*Duty;
END;
(*----- End About Limit Duty Cycle -----*)

Sum_W:=Sum_W+(W_VS+W_S+W_QS+W_SM+W_M+W_MB+W_
QB+W_B+W_VB+W_AL);
Sum_Mew:=Sum_Mew+(Mew_VS_Duty+Mew_S_Duty+Mew_QS_D
uty+Mew_SM_Duty+Mew_M_Duty

```

```

+Mew_MB_Duty+Mew_QB_Duty+Mew_B_Duty+Mew_VB_Duty+
Mew_AL_Duty);
Duty:=Sum_W/Sum_Mew; { Defuzzifion by Center of Area (COA)}
WRITELN('Sum_W = ',Sum_W;11:5,' Sum_Mew = ',Sum_Mew;11:5
);
Mew_VS_Duty:=0; Mew_S_Duty:=0; Mew_QS_Duty:=0;
Mew_SM_Duty:=0; Mew_M_Duty:=0; Mew_MB_Duty:=0;
Mew_QB_Duty:=0; Mew_B_Duty:=0; Mew_VB_Duty:=0;
Mew_AL_Duty:=0;

```

```
END; {--PROCEDURE DEFUZZIFIER--}
```

```

PROCEDURE INFERENCE_RULE_NO_LOAD
(MEW_NB_Ia,MEW_NM_Ia,MEW_NS_Ia,MEW_ZO_Ia,MEW_PS_I
a,MEW_PM_Ia,MEW_PB_Ia,MEW_NS_Sp,MEW_ZO_Sp,MEW_PS_
Sp,MEW_PM_Sp,MEW_PB_Sp:REAL;VAR Duty:REAL);
VAR (*-----Variable of Inference rule-----*)
MEW_VS1_Duty,MEW_S2_Duty,MEW_S3_Duty,MEW_VS4_Duty,
MEW_VS5_Duty,MEW_VS6_Duty,MEW_VS7_Duty,
MEW_S8_Duty,MEW_S9_Duty,MEW_SM10_Duty,MEW_SM11_Dut
y,MEW_M12_Duty,MEW_M13_Duty,MEW_M14
_Duty,MEW_QS15_Duty,MEW_SM16_Duty,MEW_M17_Duty,MEW
_MB18_Duty,MEW_QB19_Duty,MEW_B20_Duty
MEW_VB21_Duty,MEW_VS22_Duty,MEW_QS23_Duty,MEW_M24
_Duty,MEW_QB25_Duty,MEW_B26_Duty,
MEW_VB27_Duty,MEW_AL28_Duty,MEW_VS29_Duty,MEW_S30
_Duty,MEW_QS31_Duty,MEW_M32_Duty,MEW_MB33_Duty,ME
W_QB34_Duty,MEW_VB35_Duty:REAL;
Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11:REAL;
BEGIN {--BEGIN PROCEDURE--}
Z2:=0;Z3:=0;Z4:=0;Z5:=0;Z6:=0;Z7:=0;Z8:=0;Z9:=0;Z10:=0;Z11:=0;
Sum_W:=0; Sum_Mew:=0;

```

```
{----- Group of Rule 1 MEW_NS_Sp> 0 -----}
```

```
IF MEW_NS_Sp>0 THEN
```

```
BEGIN
```

```
{RULE 1.} IF (MEW_NB_Ia>0) THEN
```

```
BEGIN
```

```
WRITELN('RULE 1 ');
```

```
IF (MEW_NS_Sp<MEW_NB_Ia) THEN
```

```
BEGIN
```

```
MEW_VS1_Duty:=MEW_NS_Sp;
```

```
Z2:=MEW_VS1_Duty;
```

```
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
END ELSE
```

```
BEGIN
```

```
MEW_VS1_Duty:=MEW_NB_Ia;
```

```

      Z2:=MEW_VS1_Duty;
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {----END RULE 1----}

{RULE 2.} IF (MEW_NM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 2 ');
    IF (MEW_NS_Sp<MEW_NM_Ia) THEN
      BEGIN
        MEW_S2_Duty:=MEW_NS_Sp;
        Z3:=MEW_S2_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_S2_Duty:=MEW_ZO_Ia;
        Z3:=MEW_S2_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z3:=0;
      END; {----END RULE 2----}

{RULE 3.} IF (MEW_NS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 3 ');
    IF (MEW_NS_Sp<MEW_NS_Ia) THEN
      BEGIN
        MEW_S3_Duty:=MEW_NS_Sp;
        Z3:=MEW_S3_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_S3_Duty:=MEW_NS_Ia;
        Z3:=MEW_S3_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z3:=0;
      END; {----END RULE 3----}

{RULE 4.} IF (MEW_ZO_Ia>0) THEN
  BEGIN
    WRITELN('RULE 4 ');
    IF (MEW_NS_Sp<MEW_ZO_Ia) THEN

```

```

      BEGIN
        MEW_VS4_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS4_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS4_Duty:=MEW_ZO_Ia;
        Z2:=MEW_VS4_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {----END RULE 4----}

{RULE 5.} IF (MEW_PS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 5 ');
    IF (MEW_NS_Sp<MEW_PS_Ia) THEN
      BEGIN
        MEW_VS5_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS5_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS5_Duty:=MEW_PS_Ia;
        Z2:=MEW_VS5_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {----END RULE 5----}

{RULE 6.} IF (MEW_PM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 6 ');
    IF (MEW_NS_Sp<MEW_PM_Ia) THEN
      BEGIN
        MEW_VS6_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS6_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS6_Duty:=MEW_PM_Ia;
        Z2:=MEW_VS6_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);

```

```

END;
Z2:=0;
END; {----END RULE 6----}

{RULE 7.} IF (MEW_PB_Ia>0) THEN
BEGIN
WRITELN('RULE 7');
IF (MEW_NS_Sp<MEW_PB_Ia) THEN
BEGIN
MEW_VS7_Duty:=MEW_NS_Sp;
Z2:=MEW_VS7_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END ELSE
BEGIN
MEW_VS7_Duty:=MEW_PB_Ia;
Z2:=MEW_VS7_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z2:=0;
END; {----END RULE 7----}

END; {END GROUP OF RULE 1}

{---- Group of Rule 2 MEW_ZO_Sp> 0 ----}
IF MEW_ZO_Sp>0 THEN
BEGIN

{RULE 8.} IF (MEW_NB_Ia>0) THEN
BEGIN
WRITELN('RULE 8');
IF (MEW_ZO_Sp<MEW_NB_Ia) THEN
BEGIN
MEW_S8_Duty:=MEW_ZO_Sp;
Z3:=MEW_S8_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END ELSE
BEGIN
MEW_S8_Duty:=MEW_NB_Ia;
Z3:=MEW_S8_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z3:=0;
END; {----END RULE 8----}

{RULE 9.} IF (MEW_NM_Ia>0) THEN
BEGIN
WRITELN('RULE 9');
IF (MEW_ZO_Sp<MEW_NM_Ia) THEN
BEGIN
MEW_S9_Duty:=MEW_ZO_Sp;
Z3:=MEW_S9_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END ELSE
BEGIN
MEW_S9_Duty:=MEW_NM_Ia;
Z3:=MEW_S9_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z3:=0;
END; {----END RULE 9----}

{RULE 10.} IF (MEW_NS_Ia>0) THEN
BEGIN
WRITELN('RULE 10');
IF (MEW_ZO_Sp<MEW_NS_Ia) THEN
BEGIN
MEW_SM10_Duty:=MEW_ZO_Sp;
Z5:=MEW_SM10_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END
ELSE
BEGIN
MEW_SM10_Duty:=MEW_NS_Ia;
Z5:=MEW_SM10_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z5:=0;
END; {----END RULE 10----}

{RULE 11.} IF (MEW_ZO_Ia>0) THEN
BEGIN
WRITELN('RULE 11');
IF (MEW_ZO_Sp<MEW_ZO_Ia) THEN
BEGIN
MEW_SM11_Duty:=MEW_ZO_Sp;
Z5:=MEW_SM11_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END ELSE
BEGIN

```

```

        MEW_SM11_Duty:=MEW_ZO_Ia;
        Z5:=MEW_SM11_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
        END;
        Z5:=0;
        END; {----END RULE 11----}

{RULE 12.} IF (MEW_PS_Ia>0) THEN
        BEGIN
        WRITELN('RULE 12 ');
        IF (MEW_ZO_Sp<MEW_PS_Ia) THEN
                BEGIN
                        MEW_M12_Duty:=MEW_ZO_Sp;
                        Z6:=MEW_M12_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                        MEW_M12_Duty:=MEW_PS_Ia ;
                        Z6:=MEW_M12_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z6:=0;
        END; {----END RULE 12----}

{RULE 13.} IF (MEW_PM_Ia>0) THEN
        BEGIN
        WRITELN('RULE 13 ');
        IF (MEW_ZO_Sp<MEW_PM_Ia) THEN
                BEGIN
                        MEW_M13_Duty:=MEW_ZO_Sp;
                        Z6:=MEW_M13_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                        MEW_M13_Duty:=MEW_PM_Ia;
                        Z6:=MEW_M13_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z6:=0;
        END; {----END RULE 13----}

{RULE 14.} IF (MEW_PB_Ia>0) THEN
        BEGIN
        WRITELN('RULE 14 ');
        IF (MEW_ZO_Sp<MEW_PB_Ia) THEN
                BEGIN
                        MEW_M14_Duty:=MEW_ZO_Sp;
                        Z6:=MEW_M14_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                        MEW_M14_Duty:=MEW_PB_Ia;
                        Z6:=MEW_M14_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z6:=0;
        END; {----END RULE 14----}
        END; {END GROUP OF RULE 2}

{---- Group of Rule 3 MEW_PS_Sp> 0 ----}
IF MEW_PS_Sp>0 THEN
        BEGIN

{RULE 15.} IF (MEW_NB_Ia>0) THEN
        BEGIN
        WRITELN('RULE 15 ');
        IF (MEW_PS_Sp<MEW_NB_Ia) THEN
                BEGIN
                        MEW_QS15_Duty:=MEW_PS_Sp;
                        Z4:=MEW_QS15_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                        MEW_QS15_Duty:=MEW_NB_Ia;
                        Z4:=MEW_QS15_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z4:=0;
        END; {----END RULE 15----}

{RULE 16.} IF (MEW_NM_Ia>0) THEN
        BEGIN
        WRITELN('RULE 16 ');
        IF (MEW_PS_Sp<MEW_NM_Ia) THEN
                BEGIN
                        MEW_SM16_Duty:=MEW_PS_Sp;
                        Z5:=MEW_SM16_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);

```

```

END ELSE
BEGIN
    MEW_SM16_Duty:=MEW_NM_Ia;
    Z5:=MEW_SM16_Duty;
    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z5:=0;
END; {----END RULE 16----}

{RULE 17.} IF (MEW_NS_Ia>0) THEN
BEGIN
    WRITELN('RULE 17 ');
    IF (MEW_PS_Sp<MEW_NS_Ia) THEN
    BEGIN
        MEW_M17_Duty:=MEW_PS_Sp;
        Z6:=MEW_M17_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END
    ELSE
    BEGIN
        MEW_M17_Duty:=MEW_NS_Ia;
        Z6:=MEW_M17_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z6:=0;
END; {----END RULE 17----}

{RULE 18.} IF (MEW_ZO_Ia>0) THEN
BEGIN
    WRITELN('RULE 18 ');
    IF (MEW_PS_Sp<MEW_ZO_Ia) THEN
    BEGIN
        MEW_MB18_Duty:=MEW_PS_Sp;
        Z7:=MEW_MB18_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_MB18_Duty:=MEW_ZO_Ia;
        Z7:=MEW_MB18_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z7:=0;
END; {----END RULE 18----}

```

```

{RULE 19.} IF (MEW_PS_Ia>0) THEN
BEGIN
    WRITELN('RULE 19 ');
    IF (MEW_PS_Sp<MEW_PS_Ia) THEN
    BEGIN
        MEW_QB19_Duty:=MEW_PS_Sp;
        Z8:=MEW_QB19_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_QB19_Duty:=MEW_PS_Ia ;
        Z8:=MEW_QB19_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z8:=0;
END; {----END RULE 19----}

{RULE 20.} IF (MEW_PM_Ia>0) THEN
BEGIN
    WRITELN('RULE 20 ');
    IF (MEW_PS_Sp<MEW_PM_Ia) THEN
    BEGIN
        MEW_B20_Duty:=MEW_PS_Sp;
        Z9:=MEW_B20_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END
    ELSE
    BEGIN
        MEW_B20_Duty:=MEW_PM_Ia;
        Z9:=MEW_B20_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z9:=0;
END; {----END RULE 20----}

{RULE 21.} IF (MEW_PB_Ia>0) THEN
BEGIN
    WRITELN('RULE 21 ');
    IF (MEW_PS_Sp<MEW_PB_Ia) THEN
    BEGIN
        MEW_VB21_Duty:=MEW_PS_Sp;
        Z10:=MEW_VB21_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE

```

```

BEGIN
    MEW_VB21_Duty:=MEW_PB_Ia;
    Z10:=MEW_VB21_Duty;
    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z10:=0;
END; {----END RULE 21----}
END; {END GROUP OF RULE 3}

{---- Group of Rule 4 MEW_PM_Sp> 0 ----}
IF MEW_PM_Sp>0 THEN
    BEGIN
    {RULE 22.} IF (MEW_NB_Ia>0) THEN
        BEGIN
            WRITELN('RULE 22 ');
            IF (MEW_PM_Sp<MEW_NB_Ia) THEN
                BEGIN
                    MEW_VS22_Duty:=MEW_PM_Sp;
                    Z2:=MEW_VS22_Duty;
                    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                    MEW_VS22_Duty:=MEW_NB_Ia;
                    Z2:=MEW_VS22_Duty;
                    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z2:=0;
            END; {----END RULE 22----}

        {RULE 23.} IF (MEW_NM_Ia>0) THEN
            BEGIN
                WRITELN('RULE 23 ');
                IF (MEW_PM_Sp<MEW_NM_Ia) THEN
                    BEGIN
                        MEW_QS23_Duty:=MEW_PM_Sp;
                        Z4:=MEW_QS23_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                    END ELSE
                    BEGIN
                        MEW_QS23_Duty:=MEW_NM_Ia;
                        Z4:=MEW_QS23_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                    END;
                    Z4:=0;

```

```

END; {----END RULE 23----}

{RULE 24.} IF (MEW_NS_Ia>0) THEN
    BEGIN
        WRITELN('RULE 24 ');
        IF (MEW_PM_Sp<MEW_NS_Ia) THEN
            BEGIN
                MEW_M24_Duty:=MEW_PM_Sp;
                Z6:=MEW_M24_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
                MEW_M24_Duty:=MEW_NS_Ia;
                Z6:=MEW_M24_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z6:=0;
        END; {----END RULE 24----}

    {RULE 25.} IF (MEW_ZO_Ia>0) THEN
        BEGIN
            WRITELN('RULE 25 ');
            IF (MEW_PM_Sp<MEW_ZO_Ia) THEN
                BEGIN
                    MEW_QB25_Duty:=MEW_PM_Sp;
                    Z8:=MEW_QB25_Duty;
                    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END ELSE
                BEGIN
                    MEW_QB25_Duty:=MEW_ZO_Ia;
                    Z8:=MEW_QB25_Duty;
                    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
                END;
                Z8:=0;
            END; {----END RULE 25----}

        {RULE 26.} IF (MEW_PS_Ia>0) THEN
            BEGIN
                WRITELN('RULE 26 ');
                IF (MEW_PM_Sp<MEW_PS_Ia) THEN
                    BEGIN
                        MEW_B26_Duty:=MEW_PM_Sp;
                        Z9:=MEW_B26_Duty;
                        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);

```

```

END ELSE
BEGIN
    MEW_B26_Duty:=MEW_PS_Ia;
    Z9:=MEW_B26_Duty;
    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z9:=0;
    END; {----END RULE 26----}

{RULE 27.} IF (MEW_PM_Ia>0) THEN
BEGIN
    WRITELN('RULE 27 ');
    IF (MEW_PM_Sp<MEW_PM_Ia) THEN
    BEGIN
        MEW_VB27_Duty:=MEW_PM_Sp;
        Z10:=MEW_VB27_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_VB27_Duty:=MEW_PM_Ia;
        Z10:=MEW_VB27_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z10:=0;
    END; {----END RULE 27----}

{RULE 28.} IF (MEW_PB_Ia>0) THEN
BEGIN
    WRITELN('RULE 28 ');
    IF (MEW_PM_Sp<MEW_PB_Ia) THEN
    BEGIN
        MEW_AL28_Duty:=MEW_PM_Sp;
        Z11:=MEW_AL28_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_AL28_Duty:=MEW_PB_Ia;
        Z11:=MEW_AL28_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z11:=0;
    END; {----END RULE 28----}
END; {END GROUP OF RULE 4}

```

```

{---- Group of Rule 5 MEW_PB_Sp> 0 ----}
IF MEW_PB_Sp>0 THEN
BEGIN
    {RULE 29.} IF (MEW_NB_Ia>0) THEN
    BEGIN
        WRITELN('RULE 29 ');
        IF (MEW_PB_Sp<MEW_NB_Ia) THEN
        BEGIN
            MEW_VS29_Duty:=MEW_PB_Sp;
            Z2:=MEW_VS29_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
        END ELSE
        BEGIN
            MEW_VS29_Duty:=MEW_NB_Ia;
            Z2:=MEW_VS29_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
        END;
        Z2:=0;
        END; {----END RULE 29----}

    {RULE 30.} IF (MEW_NM_Ia>0) THEN
    BEGIN
        WRITELN('RULE 30 ');
        IF (MEW_PB_Sp<MEW_NM_Ia) THEN
        BEGIN
            MEW_S30_Duty:=MEW_PB_Sp;
            Z3:=MEW_S30_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
        END ELSE
        BEGIN
            MEW_S30_Duty:=MEW_NM_Ia;
            Z3:=MEW_S30_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
        END;
        Z3:=0;
        END; {----END RULE 30----}

    {RULE 31.} IF (MEW_NS_Ia>0) THEN
    BEGIN
        WRITELN('RULE 31 ');
        IF (MEW_PB_Sp<MEW_NS_Ia) THEN
        BEGIN
            MEW_QS31_Duty:=MEW_PB_Sp;
            Z4:=MEW_QS31_Duty;

```

```

        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_QS31_Duty:=MEW_NS_Ia;
        Z4:=MEW_QS31_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z4:=0;
    END; {----END RULE 31----}

{RULE 32.} IF (MEW_ZO_Ia>0) THEN
    BEGIN
        WRITELN('RULE 32 ');
        IF (MEW_PB_Sp<MEW_ZO_Ia) THEN
            BEGIN
                MEW_M32_Duty:=MEW_PB_Sp;
                Z6:=MEW_M32_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
                MEW_M32_Duty:=MEW_ZO_Ia;
                Z6:=MEW_M32_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z6:=0;
            END; {----END RULE 32----}

{RULE 33.} IF (MEW_PS_Ia>0) THEN
    BEGIN
        WRITELN('RULE 33 ');
        IF (MEW_PB_Sp<MEW_PS_Ia) THEN
            BEGIN
                MEW_MB33_Duty:=MEW_PB_Sp;
                Z7:=MEW_MB33_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
                MEW_MB33_Duty:=MEW_PS_Ia;
                Z7:=MEW_MB33_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z7:=0;
            END; {----END RULE 33----}

{RULE 34.} IF (MEW_PM_Ia>0) THEN
    BEGIN
        WRITELN('RULE 34 ');
        IF (MEW_PB_Sp<MEW_PM_Ia) THEN
            BEGIN
                MEW_QB34_Duty:=MEW_PB_Sp;
                Z8:=MEW_QB34_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
                MEW_QB34_Duty:=MEW_PM_Ia;
                Z8:=MEW_QB34_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z8:=0;
            END; {----END RULE 34----}

{RULE 35.} IF (MEW_PB_Ia>0) THEN
    BEGIN
        WRITELN('RULE 35 ');
        IF (MEW_PB_Sp<MEW_PB_Ia) THEN
            BEGIN
                MEW_VB35_Duty:=MEW_PB_Sp;
                Z10:=MEW_VB35_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
                MEW_VB35_Duty:=MEW_PB_Ia;
                Z10:=MEW_VB35_Duty;
                DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z10:=0;
            END; {----END RULE 35----}
    END; {END GROUP OF RULE 5}
END; {PROCEDURE INFERENCE_RULE_NO_LOAD}

PROCEDURE INFERENCE_RULE_HAVE_LOAD
(MEW_NB_Ia,MEW_NM_Ia,MEW_NS_Ia,MEW_ZO_Ia,MEW_PS_I
a,MEW_PM_Ia, MEW_PB_Ia,
MEW_NS_Sp,MEW_ZO_Sp,MEW_PS_Sp,MEW_PM_Sp,MEW_PB_
Sp:REAL;VAR Duty:REAL);
VAR
(*-----Variable of Inference rule-----*)

```

```

MEW_VS1_Duty,MEW_S2_Duty,MEW_S3_Duty,MEW_VS4_Duty,
MEW_VS5_Duty,MEW_VS6_Duty,MEW_VS7_Duty,
MEW_S8_Duty, MEW_S9_Duty,
MEW_SM10_Duty,MEW_SM11_Duty,MEW_M12_Duty,MEW_MB1
3_Duty,MEW_QB14_Duty,MEW_QS15_Duty,MEW_SM16_Duty,
MEW_M17_Duty,MEW_MB18_Duty,MEW_QB19_Duty,MEW_B20
_Duty,MEW_VB21_Duty,MEW_VS22_Duty,MEW_QS23_Duty,ME
W_M24_Duty, MEW_QB25_Duty,
MEW_B26_Duty,MEW_VB27_Duty,MEW_AL28
_Duty,MEW_VS29_Duty,MEW_S30_Duty,MEW_QS31_Duty,
MEW_M32_Duty,MEW_MB33_Duty,MEW_QB34_Duty,MEW_VB3
5_Duty:REAL;
Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11:REAL;
(*-----End Variable of Inference rule-----*)

BEGIN {--BEGIN PROCEDURE--}
Z2:=0;Z3:=0;Z4:=0;Z5:=0;Z6:=0;Z7:=0;Z8:=0;Z9:=0;Z10:=0;Z11:=0;
Sum_W:=0; Sum_Mew:=0;

{----- Group of Rule 1 MEW_NS_Sp> 0 -----}
IF MEW_NS_Sp>0 THEN
  BEGIN
    {RULE 1.} IF (MEW_NB_Ia>0) THEN
      BEGIN
        WRITELN('RULE 1 ');
        IF (MEW_NS_Sp<MEW_NB_Ia) THEN
          BEGIN
            MEW_VS1_Duty:=MEW_NS_Sp;
            Z2:=MEW_VS1_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END ELSE
          BEGIN
            MEW_VS1_Duty:=MEW_NB_Ia;
            Z2:=MEW_VS1_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END
        END;
        Z2:=0;
        END; {----END RULE 1----}

    {RULE 2.} IF (MEW_NM_Ia>0) THEN
      BEGIN
        WRITELN('RULE 2 ');
        IF (MEW_NS_Sp<MEW_NM_Ia) THEN
          BEGIN
            MEW_S2_Duty:=MEW_NS_Sp;
            Z3:=MEW_S2_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END ELSE
          BEGIN
            MEW_S2_Duty:=MEW_ZO_Ia;
            Z3:=MEW_S2_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END
        END;
        Z3:=0;
        END; {----END RULE 2----}

    {RULE 3.} IF (MEW_NS_Ia>0) THEN
      BEGIN
        WRITELN('RULE 3 ');
        IF (MEW_NS_Sp<MEW_NS_Ia) THEN
          BEGIN
            MEW_S3_Duty:=MEW_NS_Sp;
            Z3:=MEW_S3_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END ELSE
          BEGIN
            MEW_S3_Duty:=MEW_NS_Ia;
            Z3:=MEW_S3_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END
        END;
        Z3:=0;
        END; {----END RULE 3----}

    {RULE 4.} IF (MEW_ZO_Ia>0) THEN
      BEGIN
        WRITELN('RULE 4 ');
        IF (MEW_NS_Sp<MEW_ZO_Ia) THEN
          BEGIN
            MEW_VS4_Duty:=MEW_NS_Sp;
            Z2:=MEW_VS4_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END ELSE
          BEGIN
            MEW_VS4_Duty:=MEW_ZO_Ia;
            Z2:=MEW_VS4_Duty;
            DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
          END
        END;
        Z2:=0;
        END; {----END RULE 4----}

```

```

{RULE 5.} IF (MEW_PS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 5 ');
    IF (MEW_NS_Sp<MEW_PS_Ia) THEN
      BEGIN
        MEW_VS5_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS5_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS5_Duty:=MEW_PS_Ia;
        Z2:=MEW_VS5_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {---END RULE 5---}

{RULE 6.} IF (MEW_PM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 6 ');
    IF (MEW_NS_Sp<MEW_PM_Ia) THEN
      BEGIN
        MEW_VS6_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS6_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS6_Duty:=MEW_PM_Ia;
        Z2:=MEW_VS6_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {---END RULE 6---}

{RULE 7.} IF (MEW_PB_Ia>0) THEN
  BEGIN
    WRITELN('RULE 7 ');
    IF (MEW_NS_Sp<MEW_PB_Ia) THEN
      BEGIN
        MEW_VS7_Duty:=MEW_NS_Sp;
        Z2:=MEW_VS7_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VS7_Duty:=MEW_PB_Ia;
        Z2:=MEW_VS7_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z2:=0;
      END; {---END RULE 7---}

{RULE 8.} IF (MEW_NB_Ia>0) THEN
  BEGIN
    WRITELN('RULE 8 ');
    IF (MEW_ZO_Sp<MEW_NB_Ia) THEN
      BEGIN
        MEW_S8_Duty:=MEW_ZO_Sp;
        Z3:=MEW_S8_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_S8_Duty:=MEW_NB_Ia;
        Z3:=MEW_S8_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z3:=0;
      END; {---END RULE 8---}

{RULE 9.} IF (MEW_NM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 9 ');
    IF (MEW_ZO_Sp<MEW_NM_Ia) THEN
      BEGIN
        MEW_S9_Duty:=MEW_ZO_Sp;
        Z3:=MEW_S9_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_S9_Duty:=MEW_NM_Ia;
        Z3:=MEW_S9_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z3:=0;
    
```

```

END; {----END RULE 9----}

{RULE 10.} IF (MEW_NS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 10 ');
    IF (MEW_ZO_Sp<MEW_NS_Ia) THEN
      BEGIN
        MEW_SM10_Duty:=MEW_ZO_Sp;
        Z5:=MEW_SM10_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_SM10_Duty:=MEW_NS_Ia;
        Z5:=MEW_SM10_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z5:=0;
    END; {----END RULE 10----}

{RULE 11.} IF (MEW_ZO_Ia>0) THEN
  BEGIN
    WRITELN('RULE 11 ');
    IF (MEW_ZO_Sp<MEW_ZO_Ia) THEN
      BEGIN
        MEW_SM11_Duty:=MEW_ZO_Sp;
        Z5:=MEW_SM11_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_SM11_Duty:=MEW_ZO_Ia;
        Z5:=MEW_SM11_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z5:=0;
    END; {----END RULE 11----}

{RULE 12.} IF (MEW_PS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 12 ');
    IF (MEW_ZO_Sp<MEW_PS_Ia) THEN
      BEGIN
        MEW_M12_Duty:=MEW_ZO_Sp;
        Z6:=MEW_M12_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_M12_Duty:=MEW_PS_Ia;
        Z6:=MEW_M12_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z6:=0;
    END; {----END RULE 12----}

{RULE 13.} IF (MEW_PM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 13 ');
    IF (MEW_ZO_Sp<MEW_PM_Ia) THEN
      BEGIN
        MEW_MB13_Duty:=MEW_ZO_Sp;
        Z7:=MEW_MB13_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_MB13_Duty:=MEW_PM_Ia;
        Z7:=MEW_MB13_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z7:=0;
    END; {----END RULE 13----}

{RULE 14.} IF (MEW_PB_Ia>0) THEN
  BEGIN
    WRITELN('RULE 14 ');
    IF (MEW_ZO_Sp<MEW_PB_Ia) THEN
      BEGIN
        MEW_QB14_Duty:=MEW_ZO_Sp;
        Z8:=MEW_QB14_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_QB14_Duty:=MEW_PB_Ia;
        Z8:=MEW_QB14_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z8:=0;
    END; {----END RULE 14----}
  END; {END GROUP OF RULE 2}

```

```

{---- Group of Rule 3 MEW_PS_Sp> 0 ----}
IF MEW_PS_Sp>0 THEN
  BEGIN
  {RULE 15.} IF (MEW_NB_Ia>0) THEN
    BEGIN
    WRITELN('RULE 15 ');
    IF (MEW_PS_Sp<MEW_NB_Ia) THEN
      BEGIN
        MEW_QS15_Duty:=MEW_PS_Sp;
        Z4:=MEW_QS15_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_QS15_Duty:=MEW_NB_Ia;
        Z4:=MEW_QS15_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z4:=0;
    END; {----END RULE 15----}

  {RULE 16.} IF (MEW_NM_Ia>0) THEN
    BEGIN
    WRITELN('RULE 16 ');
    IF (MEW_PS_Sp<MEW_NM_Ia) THEN
      BEGIN
        MEW_SM16_Duty:=MEW_PS_Sp;
        Z5:=MEW_SM16_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_SM16_Duty:=MEW_NM_Ia;
        Z5:=MEW_SM16_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z5:=0;
    END; {----END RULE 16----}

  {RULE 17.} IF (MEW_NS_Ia>0) THEN
    BEGIN
    WRITELN('RULE 17 ');
    IF (MEW_PS_Sp<MEW_NS_Ia) THEN
      BEGIN
        MEW_M17_Duty:=MEW_PS_Sp;
        Z6:=MEW_M17_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_M17_Duty:=MEW_NS_Ia;
        Z6:=MEW_M17_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z6:=0;
    END; {----END RULE 17----}

  {RULE 18.} IF (MEW_ZO_Ia>0) THEN
    BEGIN
    WRITELN('RULE 18 ');
    IF (MEW_PS_Sp<MEW_ZO_Ia) THEN
      BEGIN
        MEW_MB18_Duty:=MEW_PS_Sp;
        Z7:=MEW_MB18_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_MB18_Duty:=MEW_ZO_Ia;
        Z7:=MEW_MB18_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z7:=0;
    END; {----END RULE 18----}

  {RULE 19.} IF (MEW_PS_Ia>0) THEN
    BEGIN
    WRITELN('RULE 19 ');
    IF (MEW_PS_Sp<MEW_PS_Ia) THEN
      BEGIN
        MEW_QB19_Duty:=MEW_PS_Sp;
        Z8:=MEW_QB19_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_QB19_Duty:=MEW_PS_Ia;
        Z8:=MEW_QB19_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z8:=0;
    END; {----END RULE 19----}

```

```
{RULE 20.} IF (MEW_PM_Ia>0) THEN
```

```
BEGIN
```

```
  WRITELN('RULE 20 ');
```

```
  IF (MEW_PS_Sp<MEW_PM_Ia) THEN
```

```
    BEGIN
```

```
      MEW_B20_Duty:=MEW_PS_Sp;
```

```
      Z9:=MEW_B20_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END ELSE
```

```
    BEGIN
```

```
      MEW_B20_Duty:=MEW_PM_Ia;
```

```
      Z9:=MEW_B20_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END;
```

```
  Z9:=0;
```

```
END; {---END RULE 20---}
```

```
{RULE 21.} IF (MEW_PB_Ia>0) THEN
```

```
BEGIN
```

```
  WRITELN('RULE 21 ');
```

```
  IF (MEW_PS_Sp<MEW_PB_Ia) THEN
```

```
    BEGIN
```

```
      MEW_VB21_Duty:=MEW_PS_Sp;
```

```
      Z10:=MEW_VB21_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END ELSE
```

```
    BEGIN
```

```
      MEW_VB21_Duty:=MEW_PB_Ia ;
```

```
      Z10:=MEW_VB21_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END;
```

```
  Z10:=0;
```

```
END; {---END RULE 21---}
```

```
END; {END GROUP OF RULE 3}
```

```
{---- Group of Rule 4 MEW_PM_Sp> 0 ----}
```

```
IF MEW_PM_Sp>0 THEN
```

```
  BEGIN
```

```
{RULE 22.} IF (MEW_NB_Ia>0) THEN
```

```
  BEGIN
```

```
    WRITELN('RULE 22 ');
```

```
    IF (MEW_PM_Sp<MEW_NB_Ia) THEN
```

```
      BEGIN
```

```
        MEW_VS22_Duty:=MEW_PM_Sp;
```

```
        Z2:=MEW_VS22_Duty;
```

```
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
      END ELSE
```

```
      BEGIN
```

```
        MEW_VS22_Duty:=MEW_NB_Ia;
```

```
        Z2:=MEW_VS22_Duty;
```

```
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
      END;
```

```
      Z2:=0;
```

```
END; {---END RULE 22---}
```

```
{RULE 23.} IF (MEW_NM_Ia>0) THEN
```

```
BEGIN
```

```
  WRITELN('RULE 23 ');
```

```
  IF (MEW_PM_Sp<MEW_NM_Ia) THEN
```

```
    BEGIN
```

```
      MEW_QS23_Duty:=MEW_PM_Sp;
```

```
      Z4:=MEW_QS23_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END ELSE
```

```
    BEGIN
```

```
      MEW_QS23_Duty:=MEW_NM_Ia;
```

```
      Z4:=MEW_QS23_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END;
```

```
      Z4:=0;
```

```
END; {---END RULE 23---}
```

```
{RULE 24.} IF (MEW_NS_Ia>0) THEN
```

```
BEGIN
```

```
  WRITELN('RULE 24 ');
```

```
  IF (MEW_PM_Sp<MEW_NS_Ia) THEN
```

```
    BEGIN
```

```
      MEW_M24_Duty:=MEW_PM_Sp;
```

```
      Z6:=MEW_M24_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END ELSE
```

```
    BEGIN
```

```
      MEW_M24_Duty:=MEW_NS_Ia;
```

```
      Z6:=MEW_M24_Duty;
```

```
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
```

```
Z8,Z9,Z10,Z11,Duty);
```

```
    END;
```

```
      Z6:=0;
```

```
END; {---END RULE 24---}
```

```
{RULE 25.} IF (MEW_ZO_Ia>0) THEN
```

```

BEGIN
  WRITELN('RULE 25 ');
  IF (MEW_PM_Sp<MEW_ZO_Ia) THEN
    BEGIN
      MEW_QB25_Duty:=MEW_PM_Sp;
      Z8:=MEW_QB25_Duty;
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
      MEW_QB25_Duty:=MEW_ZO_Ia;
      Z8:=MEW_QB25_Duty;
      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z8:=0;
  END; {----END RULE 25----}
{RULE 26.} IF (MEW_PS_Ia>0) THEN
  BEGIN
    WRITELN('RULE 26 ');
    IF (MEW_PM_Sp<MEW_PS_Ia) THEN
      BEGIN
        MEW_B26_Duty:=MEW_PM_Sp;
        Z9:=MEW_B26_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_B26_Duty:=MEW_PS_Ia;
        Z9:=MEW_B26_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z9:=0;
    END; {----END RULE 26----}
{RULE 27.} IF (MEW_PM_Ia>0) THEN
  BEGIN
    WRITELN('RULE 27 ');
    IF (MEW_PM_Sp<MEW_PM_Ia) THEN
      BEGIN
        MEW_VB27_Duty:=MEW_PM_Sp;
        Z10:=MEW_VB27_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_VB27_Duty:=MEW_PM_Ia;
        Z10:=MEW_VB27_Duty;

```

```

      DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z10:=0;
  END; {----END RULE 27----}
{RULE 28.} IF (MEW_PB_Ia>0) THEN
  BEGIN
    WRITELN('RULE 28 ');
    IF (MEW_PM_Sp<MEW_PB_Ia) THEN
      BEGIN
        MEW_AL28_Duty:=MEW_PM_Sp;
        Z11:=MEW_AL28_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END ELSE
      BEGIN
        MEW_AL28_Duty:=MEW_PB_Ia;
        Z11:=MEW_AL28_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
      END;
      Z11:=0;
    END; {----END RULE 28----}
  END; {END GROUP OF RULE 4}
  {----- Group of Rule 5 MEW_PB_Sp> 0 -----}
  IF MEW_PB_Sp>0 THEN
    BEGIN
      {RULE 29.} IF (MEW_NB_Ia>0) THEN
        BEGIN
          WRITELN('RULE 29 ');
          IF (MEW_PB_Sp<MEW_NB_Ia) THEN
            BEGIN
              MEW_VS29_Duty:=MEW_PB_Sp;
              Z2:=MEW_VS29_Duty;
              DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END ELSE
            BEGIN
              MEW_VS29_Duty:=MEW_NB_Ia;
              Z2:=MEW_VS29_Duty;
              DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
            END;
            Z2:=0;
          END; {----END RULE 29----}
        {RULE 30.} IF (MEW_NM_Ia>0) THEN
          BEGIN
            WRITELN('RULE 30 ');

```

```

IF (MEW_PB_Sp<MEW_NM_Ia) THEN
BEGIN
    MEW_S30_Duty:=MEW_PB_Sp;
    Z3:=MEW_S30_Duty;
    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END ELSE
BEGIN
    MEW_S30_Duty:=MEW_NM_Ia;
    Z3:=MEW_S30_Duty;
    DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z3:=0;
END; {----END RULE 30----}
{RULE 31.} IF (MEW_NS_Ia>0) THEN
BEGIN
    WRITELN('RULE 31 ');
    IF (MEW_PB_Sp<MEW_NS_Ia) THEN
    BEGIN
        MEW_QS31_Duty:=MEW_PB_Sp;
        Z4:=MEW_QS31_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_QS31_Duty:=MEW_NS_Ia;
        Z4:=MEW_QS31_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z4:=0;
END; {----END RULE 31----}
{RULE 32.} IF (MEW_ZO_Ia>0) THEN
BEGIN
    WRITELN('RULE 32 ');
    IF (MEW_PB_Sp<MEW_ZO_Ia) THEN
    BEGIN
        MEW_M32_Duty:=MEW_PB_Sp;
        Z6:=MEW_M32_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END
    ELSE BEGIN
        MEW_M32_Duty:=MEW_ZO_Ia;
        Z6:=MEW_M32_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
END;

```

```

Z6:=0;
END; {----END RULE 32----}
{RULE 33.} IF (MEW_PS_Ia>0) THEN
BEGIN
    WRITELN('RULE 33 ');
    IF (MEW_PB_Sp<MEW_PS_Ia) THEN
    BEGIN
        MEW_MB33_Duty:=MEW_PB_Sp;
        Z7:=MEW_MB33_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_MB33_Duty:=MEW_PS_Ia;
        Z7:=MEW_MB33_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z7:=0;
END; {----END RULE 33----}
{RULE 34.} IF (MEW_PM_Ia>0) THEN
BEGIN
    WRITELN('RULE 34 ');
    IF (MEW_PB_Sp<MEW_PM_Ia) THEN
    BEGIN
        MEW_QB34_Duty:=MEW_PB_Sp;
        Z8:=MEW_QB34_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN
        MEW_QB34_Duty:=MEW_PM_Ia;
        Z8:=MEW_QB34_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END;
    Z8:=0;
END; {----END RULE 34----}
{RULE 35.} IF (MEW_PB_Ia>0) THEN
BEGIN
    WRITELN('RULE 35 ');
    IF (MEW_PB_Sp<MEW_PB_Ia) THEN
    BEGIN
        MEW_VB35_Duty:=MEW_PB_Sp;
        Z10:=MEW_VB35_Duty;
        DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
    END ELSE
    BEGIN

```

```

MEW_VB35_Duty:=MEW_PB_Ia;
Z10:=MEW_VB35_Duty;
DEFUZZI_DUTY(Z2,Z3,Z4,Z5,Z6,Z7,
Z8,Z9,Z10,Z11,Duty);
END;
Z10:=0;
END; {----END RULE 35----}

END; {END GROUP OF RULE 5}
END; {PROCEDURE INFERENCE_RULE_HAVE_LOAD}
BEGIN {main program}
  CLRSCR;
  FILENAME1:='j6.DAT'; {data file}
  WRITELN('WRITING DATA FILE : ',FILENAME1);
  ASSIGN(FILVAR1,FILENAME1); {assign filename}
  REWRITE(FILVAR1); {construct and open file}
  (*-----*)
  Ra:=1.0; {Ra=armature resistor(ohm)} La:=0.046;
  {La=armature inductor(henry)}
  kb:=0.55; {kb=back emf constant(V.sec/rad)} kt:=0.55;
  {kt=torque constant(N.m./A)}
  J:=0.093; {J=moment of inertia(kg-m^2)} B:=0.008;
  {B=viscous friction(N.m.sec/rad)}
  TL:=0.0; {TL=load torque(N-m)}
  (*-----*)
  T:=0; X1:=0; X2:=0; X3:=0; X4:=0;
  H:=Time_Period1/50;
  Ia_ref:=13; Sp_ref:=100*0.057;
  Time_Period:=Time_Period1;
  Sum_Ia:=0; Sum_Td:=0; Sum_Sp:=0; Duty:=0.0;
  Ton:=Time_Period*Duty;
  REPEAT {-----BEGIN LOOP1-----}
    (*----- Command set of chopper slowly add Ton-----*)
    (*----- Armature side and Field side -----*)
    IF (T<=Ton) OR ((T-Ton) < (H/10)) THEN {----duty interval----}
      BEGIN
        IF T>Time_Period1 THEN V:=110.;
        END;

        IF (Ton<T) AND (T<Time_Period) AND ((T-Ton)>(H/10))
        THEN
          {----freewheeling interval----}
          BEGIN V:=0.0; END;

        IF (T>=Time_Period) OR (AQS(T-Time_Period)<(H/10)) THEN
          BEGIN
            Ia:=Sum_Ia/Num_X; {Ia=average current}
            Td:=Sum_Td/Num_X; {Td=average developed torque}
            Sp:=Sum_Sp/Num_X; {Sp=average speed motor}

```

```

Sum_Ia:=0; Sum_Td:=0; Sum_Sp:=0;
Time_Period_Old:=Time_Period;
Time_Period_New:=Time_Period_Old+Time_Period1;
Time_Period:=Time_Period_New;
Ia_Tran:=0.5*Ia; {Ia through current transducer}
Sp_Tran:=0.057*Sp*(1-exp(-T/0.1));
{Sp_Tran is Sp pass Tachogenerator kt=0.057 V/A
and tachogenerator filter time constance =0.1}
E_Ia:=Ia_ref-Ia_Tran; E_Sp:=Sp_ref-Sp_Tran;
(*-----FLC Controller -----*)
GOTOXY(1,2);WRITE(' Sp_ref = ',Sp_ref:10:4,' V. ');
GOTOXY(1,3);WRITE(' Sp_Tran = ',Sp_Tran:10:4,' V. ');
GOTOXY(1,4);WRITE(' Sp = ',Sp:10:4,' V. '); GOTOXY
(1,5);WRITE(' E_Sp = ',E_Sp:10:4,' V. ');
GOTOXY(40,2);WRITE(' Ia_ref = ',Ia_ref:10:4,' V. ');
GOTOXY(40,3);WRITE(' Ia_Tran = ',Ia_Tran:10:4,' V. ');
GOTOXY(40,4);WRITE(' Ia = ',Ia:10:4,' V. '); GOTOXY
(40,5);WRITE(' E_Ia = ',E_Ia:10:4,' V. ');

MEMBERSHIP_Ia(E_Ia,MEW_NB_Ia,MEW_NM_Ia,MEW_NS_Ia,M
EW_ZO_Ia,MEW_PS_Ia,MEW_PM_Ia,MEW_PB_Ia);
{FUZZIFICATION PROCESS}

MEMBERSHIP_Sp(E_Sp,MEW_NS_Sp,MEW_ZO_Sp,MEW_PS_Sp,
MEW_PM_Sp,MEW_PB_Sp); {FUZZIFICATION PROCESS}
WRITELN;WRITELN;WRITELN;
IF TL<=0 THEN
  BEGIN

    INFERENCE_RULE_NO_LOAD(MEW_NB_Ia,MEW_NM_Ia,MEW
    _NS_Ia,MEW_ZO_Ia
    ,MEW_PS_Ia,MEW_PM_Ia,MEW_PB_Ia,MEW_NS_Sp,MEW_ZO_S
    p
    ,MEW_PS_Sp,MEW_PM_Sp,MEW_PB_Sp,Duty);
    END ELSE
    BEGIN

    INFERENCE_RULE_HAVE_LOAD(MEW_NB_Ia,MEW_NM_Ia,M
    EW_NS_Ia,MEW_ZO_Ia
    ,MEW_PS_Ia,MEW_PM_Ia,MEW_PB_Ia,MEW_NS_Sp,MEW_ZO_S
    p
    ,MEW_PS_Sp,MEW_PM_Sp,MEW_PB_Sp,Duty);
    END;
  WRITELN;
  WRITELN('Duty = ',Duty:8:4);
  Pulse_Width_Ton_New:=Duty*Time_Period1;
  Pulse_Width_Ton:=Pulse_Width_Ton_New;

```

```

Ton_New:=Time_Period_Old+Pulse_Width_Ton;
Ton:=Ton_New;
IF T>Time_Period1 THEN V:=110;
Counter_V:=0;

(*----- Over current protection-----*)
IF Ia_Tran > 0.5*40 THEN
  BEGIN
    Ton:=Time_Period_Old;
    WRITELN('Over current Ia = ',Ia:8:4);
    Sound(220); { Beep } Delay(200); { For 200 ms }
    NoSound; { Relief! } { Halt(1); { Halt right here! } }
  END;
END;

IF T>3 THEN TL:=9; { Step load torque (TL) is applied at T=3.0
SEC}

(*----- Over current protection-----*)
IF Ia_Tran > 0.5*40 THEN BEGIN V:=0; END;
{Ia_Maximum be 2 times the rated current }
(*-----Set command calculation Fourth-order runge-kutta method---*)
K1:=H*FUNCX1D(T,X1,X2,X3,Ra,La,kb,V);
L1:=H*FUNCX2D(T,X1,X2,X3,kt,J,B,TL);
M1:=H*FUNCX3D(T,X1,X2,X3);
K2:=H*FUNCX1D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1,Ra,La,kb,V);
L2:=H*FUNCX2D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1,kt,J,B,TL);
M2:=H*FUNCX3D(T+0.5*H,X1+0.5*K1,X2+0.5*
L1,X3+0.5*M1);
K3:=H*FUNCX1D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2,Ra,La,kb,V);
L3:=H*FUNCX2D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2,kt,J,B,TL);
M3:=H*FUNCX3D(T+0.5*H,X1+0.5*K2,X2+0.5*
L2,X3+0.5*M2);
K4:=H*FUNCX1D(T+H,X1+K3,X2+L3,X3+M3,Ra,La,kb,V);
L4:=H*FUNCX2D(T+H,X1+K3,X2+L3,X3+M3,kt,J,B,TL);
M4:=H*FUNCX3D(T+H,X1+K3,X2+L3,X3+M3);
X1:=X1+(K1+2*K2+2*K3+K4)/6;
X2:=X2+(L1+2*L2+2*L3+L4)/6;
X3:=X3+(M1+2*M2+2*M3+M4)/6;

IF X1<=0 THEN X1:=0;

(*---Compute Developed Torque-----*)
Td:=kt*X1; {Td=Develop torque}
(*-----*)

```

```

(*-----Compute Average Armature Current:Ia,Average Speed,Average
Torque-----*)
IF (T <=Time_Period) OR (AQS(T-Time_Period)<(H/10)) THEN
  BEGIN
    Sum_Ia:= X1+Sum_Ia;
    Sum_Td:= Td+Sum_Td;
    Sum_Sp:= X2+Sum_Sp;
    Num_X:=Time_Period1/H;
  END; (*-----end compute -----*)
(*-----Compute Vt-----*)
IF V=110 THEN
  BEGIN
    Counter_V:=Counter_V+1;
  END
  ELSE
  BEGIN
    Duty_Cycle_Vt:=((Counter_V-1)*H)/Time_Period1;
    IF T>Time_Period1 THEN
      BEGIN
        Vt:=Duty_Cycle_Vt*110;
        IF Vt<0 THEN Vt:=0; {Vt=Terminal voltage or average
voltage}
      END
    END; (*-----end compute Vt-----*)

    IF (T>=0.0) AND (T<5.0) THEN {Given print follow condition
IF statement }
      BEGIN
        IF T>=T_Print THEN BEGIN
          WRITELN(FILVAR1,T:11:5,Vt:10:4,Duty:10:4,Ia:11:4,Sp:11:4);
          T_Print:=T_Print+Time_Period1; END;
        END;

        GOTOXY(45,14);WRITE(' T =',T:7:5,' sec. ');
        GOTOXY(45,15);WRITE(' Duty Cycle =',Duty:7:4,' sec. ');
        GOTOXY(45,16);WRITE('Armature voltage:Vt =',Vt:7:4,' V. ');
        GOTOXY(45,17);WRITE(' Counter_V =',Counter_V:5,' A. ');
        GOTOXY(45,18);WRITE(' Average Current =',Ia:8:4,' A. ');
        GOTOXY(40,19);WRITE(' Actual Current =',X1:8:4,' A. ');
        GOTOXY(45,20);WRITE(' Angular velocity =',X2:8:4,' r/s ');
        GOTOXY(45,21);WRITELN('Angular displacement =',X3:8:4,' rad ');
        T:=T+H;
        UNTIL T>=Time_End;
        {-----}
        CLOSE(FILVAR1);
      END. {end program}

```