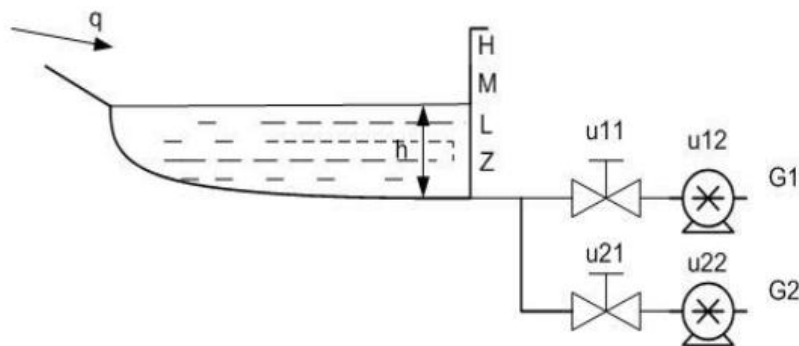


Student: Diana Elena Niti

Specializare: IAISC an 1

Problema:



Cerinte

Se da structura simplificata a unei centrale hidro-electrice avand urmatoarele caracteristici:

- este echipata cu doua generatoare $G1$ si $G2$;
- fluxul de apa de intrare este data de variabila ' q ';
- senzorii Z (zero), L (low), M (medium) si H (high) semnalizeaza cand lacul atinge nivelul lor;
- sistemul de control primeste de la senzori informatia de nivel si trebuie sa controleze generatoarele $G1$ si $G2$ prin intermediul semnalelor de control u_{11} , u_{12} , u_{21} si u_{22} astfel incat nivelul de referinta dorit sa fie mentinut;

Sistemul de control va actiona conform regulilor:

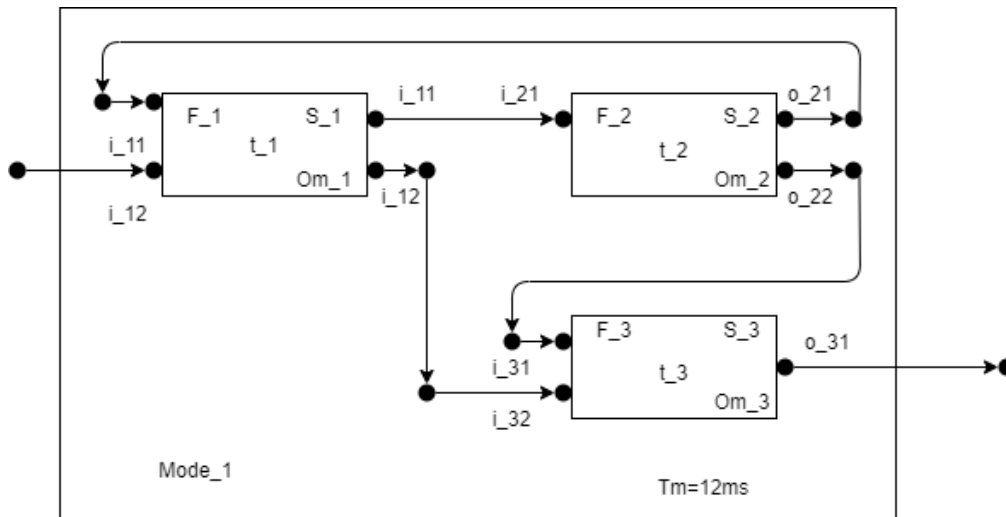
- cand nivelul este M start $G1$;
- cand nivelul este H start $G2$;
- cand nivelul este L stop $G2$;
- cand nivelul este Z stop $G1$;

Sistemul de control este compus din:

- 1 controler cu evenimente discrete (DEC) pentru pornirea si oprirea generatoarelor conform regulilor de mai sus;
- 2 controlere cu timp discret (DTC), cate unul pentru fiecare generator, pentru controlul iesirii generatoarelor;

Sa se realizeze diagrama Giotto pentru sistemul descris.

Rezolvare:

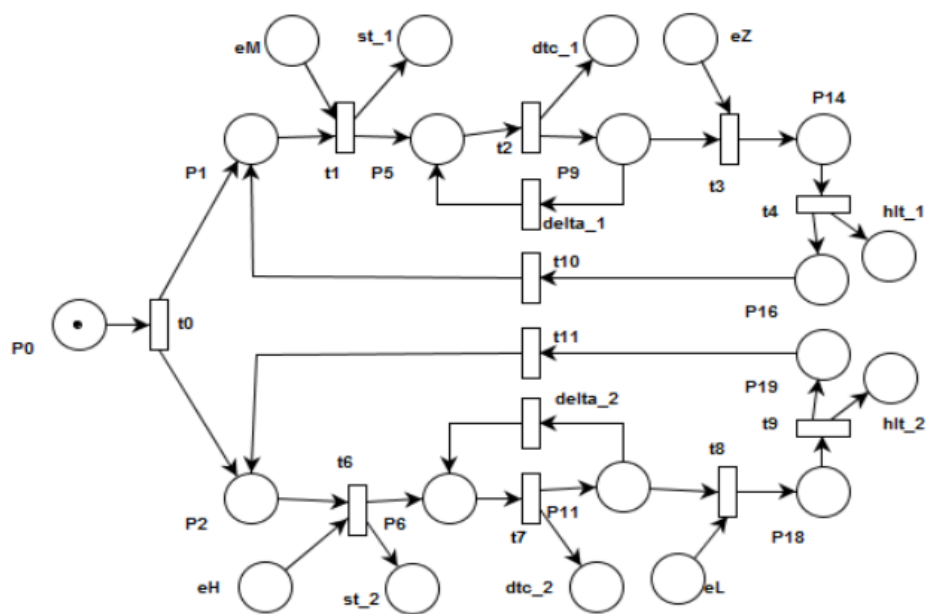
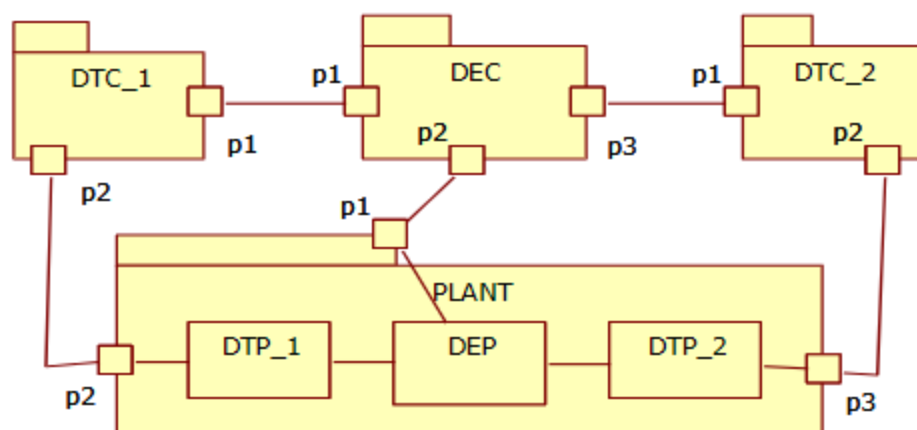


Planificarea: Om_1=6 ms Om_2=4 ms Om_3=3 ms Perioada: Tm=12 ms.

Proiectarea și implementarea softwareului: metoda GIOTTO
 Conceperea sistemului de control: rețele Petri (cu logică fuzzy)
 Implementarea aplicației: limbajul Java

Modelarea aplicației:

- Modelele instalației: DEP, DTP_1, DTP_2
- Modelele controllerelor DEC, DTP



DEC= DE Controller<- pornește și oprește controllerele DTC
DTC= DT Controller

Proiectarea , integrarea modulelor in arhitectura.

Moduri :

- Pornire → controllere implicate DEC
- Generare_1 → DEC + DTC_1
- Generare_1&2 → DEC + DTC_1 + DTC_2

Evenimente de intrare: eH, eM, eL, eZ.

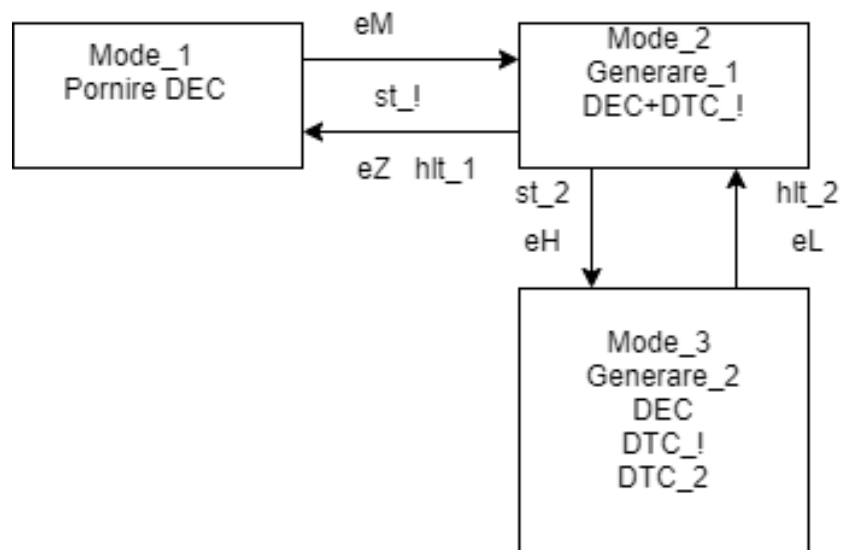
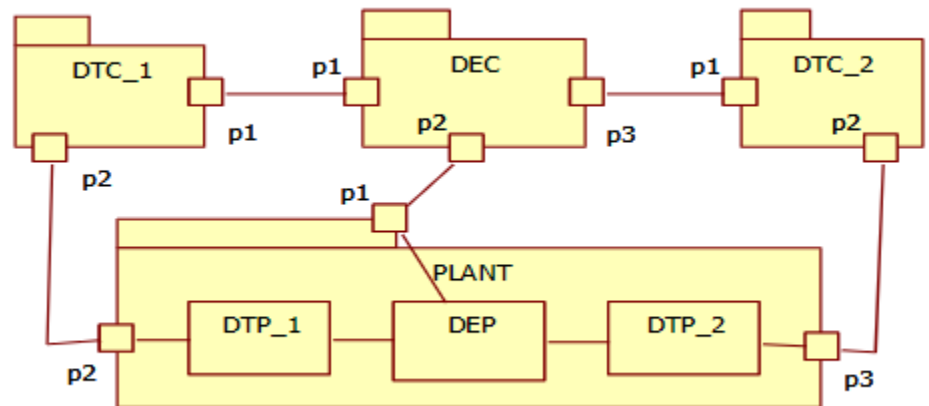
Switch Mode events
(determină schimbarea
modurilor)•

When ↑eM start G1

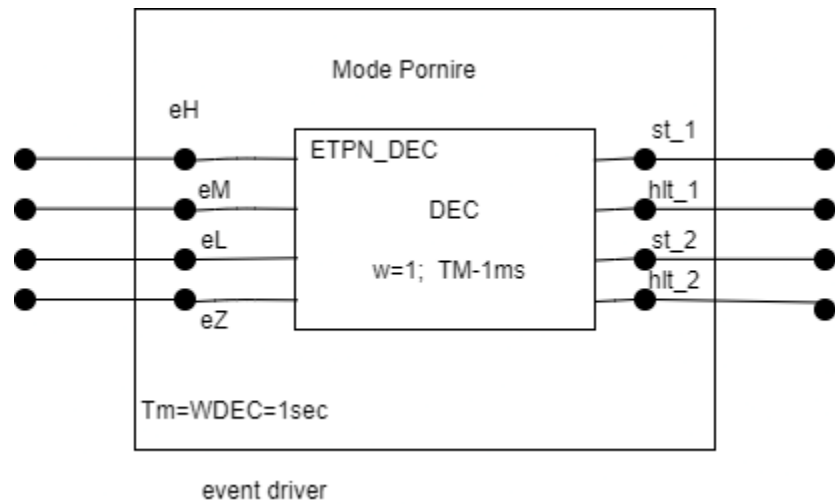
When ↑eH start G2

When ↓eL halt G2

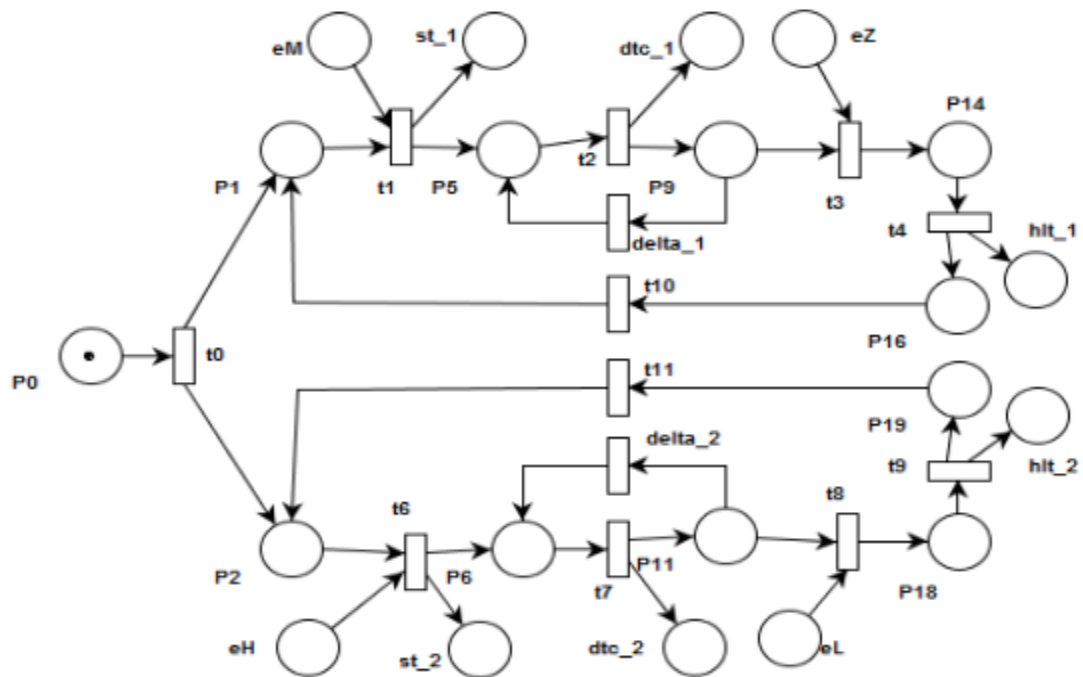
When ↓eZ halt G1



Mode Pornire:



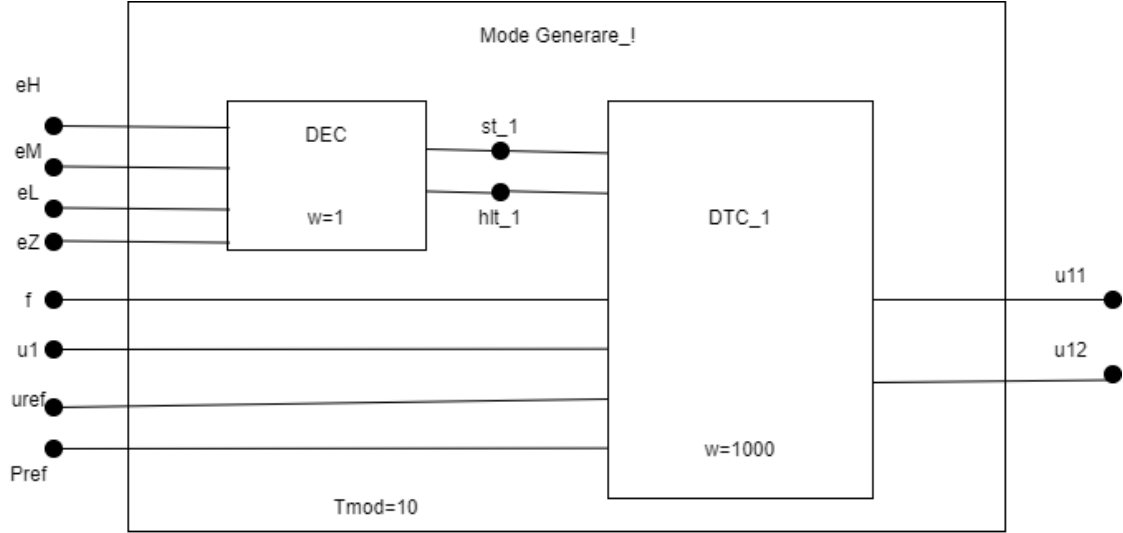
Stări interne:marcajul rețelei



Mode Generare_1

Model

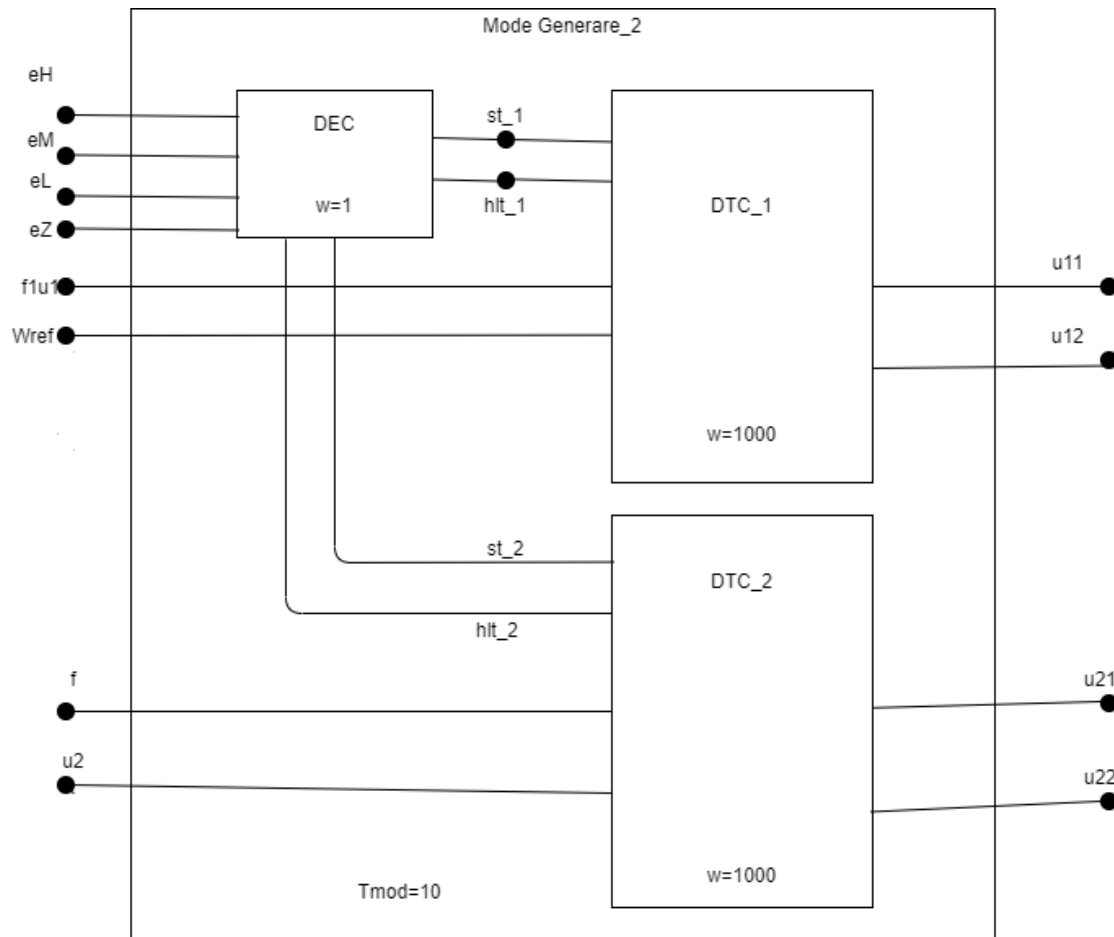
Stari interne



Mode Generare_2

Model

Stari interne



Sistemul de control al lacului:

```
public class Generator {
    Lake lake;
    double x1=0.5;// frequency variation
    double x2=0.5; //rotor angular position
    double x3=0.5; // flux variation
    double u1=0; // torque variation
    double u2=0; // rotor voltage
    double y1;//rotor angular speed
    double y2; //terminal voltage variation
    double a11=-0.1;
```

```

double a12=-0.2;
double a13=-0.1;
double a21=-0.2;
double a32=-0.1;
double a33=01;
double b11=0.2;
double b32=0.1;
double c11=0.2;
double c12=0.1;
double c13=0.1;
double c21=0.2;
double c22=0.1;
double c23=0.1;
double d11=0.2;
double d12=0.1;
double d21=0.5;
double d22=0.1;

double c1; //torque control signal u1= c1*lake.u1
double c2; //voltage control signal u2 = c2

public boolean genStarted=false; //control start&stop

public Generator (){
}public void step() {
    System.out.println("Generator started");
    for (int i=0; i<2000;i++) { //execute 2000 times generator simulation
        if (!genStarted) {u1=0; u2=0;}
        else {u1=c1; u2=c2;}
        x1 = a11*x1 + a12*x2 + a13*x3 +b11*u1;
        x2 = a21*x2;
    }
}

```



```
x3 = a32*x2 + a33*x3 + b32*u2;
```

```
y1= c11*x1+c12*x2+c13*x3 + d11*u1 + d12*u2;
```

```
y2= c21*x1+c22*x2+c23*x3 + d21*u1 + d22*u2;
```

```
System.out.println("Generator: " + " u1= "+u1+ "; u2=" + u2+"; y1= "+y1+"; y2="
```

```
+y2);
```

```
}
```

```
}
```

```
}
```