

객체지향기법을 적용한 디지털 컴퓨터  
시뮬레이터를 이용한 동기 발전기 시뮬레이션

박지호\* · 백영식  
경북대학교 전기공학과

Simulation of Synchronous Machines  
Using Object-Oriented Digital Computer Simulator

Ji-Ho Park · Young-Sik Baek  
Dept. of Electrical. Eng. Kyung Pook Univ.

Abstract

In power system stability analysis, modelling of the synchronous machine is necessary and very important. In this paper, a synchronous machine is modeled and simulated by using Object-Oriented method. The mathematical equations describing the dynamic behavior of the synchronous machine is represented by block diagram and Objected-Oriented Digital Computer Simulator(ODCS). The developed method is tested for a one-machine-to-infinite-bus system, which is accurate and very useful for a multi-machine system simulation.

1. 서론

동기기의 시뮬레이션은 아날로그 컴퓨터 또는 디지털 컴퓨터에 의한 방법의 두가지가 있다. 동기기를 묘사하는 방정식이 비선형이고 여자 시스템의 리미터와 자기회로의 포화현상도 비선형 요소이다. 이러한 비선형성은 아날로그 컴퓨터에 의해 효과적으로 실현될 수 있지만 아날로그 컴퓨터는 가격이 고가이고 그 구성이 복잡하고 어렵다. 반면 디지털 컴퓨터는 계속적인 가격 하락과 더불어 그 사용이 일반화 되었고 ACSL 이나 SIMULINK 와 같은 상용화된 패키지가 여러 분야에서 사용되고 있다. 하지만 이들 패키지 역시 고가이고 그 내용이 비공개이다. 본 논문은 객체지향 기법을 이용하여 디지털 컴퓨터에서 아날로그 컴퓨터를 실현하고 이것을 동기기의 시뮬레이션에 적용하였다. 객체지향기법을 사용한 디지털 컴퓨터 시뮬레이터(ODCS)는 미분방정식의 해법이 필요한 모든 문제에 사용할 수 있고, 아날로그 컴퓨터의 구성과 마찬가지로 미리 만들어진 객체를 프로그래밍 과정에서 사용하므로 프로그램이 쉽고 간단하다. ODCS를 사용하여 동기 발전기를 모델링하고 무한대 모션에 연결하여 시뮬레이션하고 기존의 방법과 비교하였다.

2. 객체지향 프로그래밍(OOP)

객체지향 프로그래밍을 간단히 말하면 객체를 사용한 프로그래밍이라 할 수 있다. 객체란 대상시스템의 정보를 나타내는 데이터와 그 데이터에 적용되는 함수를 결합시킨 언어구조이다. OOP 에서 객체가 가장 중요하고 함수는 이차적인 문제이다. OOP의 특징으로는 캡슐화, 계승, 다형성등이 있다. 이러한 성질을 이용하면 하나의 객체가 보다 복잡한 객체를 만드는 단위로 사용될 수 있고, 이는 저항, 콘덴서, 스위치가 전기회로를 구성하는 것과 같은 개념이다. 객체지향 프로그래밍은 프로그램을 보다 가시화 시켜준다.

3. 객체지향기법을 이용한 디지털 컴퓨터 시뮬레이터(ODCS)

아날로그 컴퓨터의 기본소자들은 덧셈기, 적분기, 분압계, 함수 발생기, 증폭기, 곱셈기 등이 있으며 이것은 그림 1 에 나타냈다.

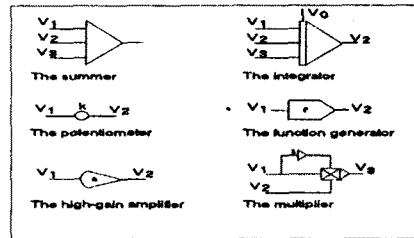


그림 1 아날로그 컴퓨터 기본소자

ODCS의 기본소자들은 이들 아날로그 컴퓨터의 기본소자들의 역할을 그대로 객체화한 것이다. 이들 소자들은 다음의 플래스이다. 적분기는 수치적으로 안정한 특성을 가지는 사다리꼴 방식을 사용하여 구성하였다[1][2].

- 1) class Summation
- 2) class Gain
- 3) class Multiply
- 4) class Integrator ; 1/s
- 5) class Transfer ; (As+B)/(Cs+D)

Transfer 플래스는 전분기 클래스에서 파생된 것이고 분모의 차수가 2차 이상인 전달함수도 구성할 수 있다.

4. 동기기의 모델링

4.1 Park 의 변환을 이용한 동기기의 일반적 모델링

동기기의 모델링의 일반적 방법은 Park 의 변환을 이용하여 고정자 변수를 회전자 기준속도로 변환해서 등가회로( d-q 등가회로)를 구성하고, 단위값으로 변환하는 것이다. Park 변환은 3 상의 시변 변수를 사분변의 일정변수로 바꾸어주고 단위값으로의 변환은 각 변수들을 정규화 시켜주어 시스템의 해석을 용이하게 해준다. 그림 2 는 동기 발전기의 d-q 축 등가회로이다.

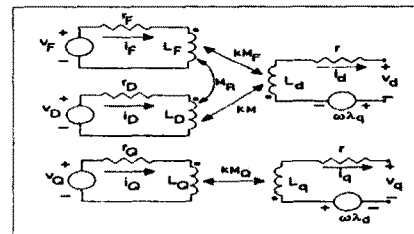


그림 2 동기발전기 d-q축 등가회로

영상 전압  $v_0$  는 3상 평형 시스템을 가정하므로 무시한다.

4.2 과도 등가회로

과도 등가회로는 여러가지가 있지만 본 논문에서 사용한 모델은 이축 등가회로이다[3]. 대규모 계통의 안정도 해석에는 기본적으로 두가지 가정을 한다.  $\lambda_d = \lambda_q = 0, \omega = 1.0$  pu 즉 고정자 축의 과도형상을 무시하고, 주파수가 일정하다고 가정한다. 이러한 가정은 동기발전기를 계통에 연결했을때 고정자에는 기본 주파수 성분만을 고려하게되므로 고정자측에는 전압방정식이 정상상태의 대수방정식으로 주어지고 정상상태로의 해석이 가능하게된다. 고주파 성분이 포함되면 수치적분시 많은 시간이 걸리게되고 계통의 규모가 커지면 계산은 불가능하게된다[4]. 발전기의 과도 등가회로는 그림 3 에 나타났다.

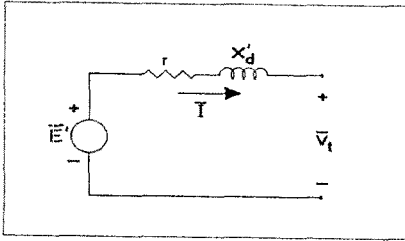


그림 3 동기발전기 과도 등가회로

과도 상태의 전압방정식, 토크방정식 그리고 동요방정식은 다음과 같다.

$$V_d = -rI_d - X'_d I_q + E'_d \quad (1)$$

$$V_q = -rI_q + X'_d I_d + E'_q \quad (2)$$

$$T'_d \dot{E}'_d = -E'_d - (X_q - X'_q) I_q \quad (3)$$

$$T'_q \dot{E}'_q = (E_{FD} - E) \quad (4)$$

$$E = E'_q + (X'_d - X_d) I_d \quad (5)$$

$$T_e = E'_d I_d + E'_q I_q - (L'_q - L'_d) I_d I_q \quad (6)$$

$$T_r \dot{\omega} = T_m - D\omega - T_e \quad (7)$$

$$\delta = \omega - 1 \quad (8)$$

### 5. 1기 무한대 계통에 적용

무한대 모선에 연결된 1기의 동기 발전기에 대하여 과도 상태를 ODCS를 이용하여 해석한다. ODCS를 이용하기 위해서 관련 방정식을 블럭선도로 나타내고 이들을 객체화하여 각 객체의 연결로서 해를 구한다.

#### 5.1 무한대 모선에 연결된 동기 발전기.

동기 발전기? 무한대 모선에 연결된 경우 과도해석은 발전기 과도모델에 무한대 모선을 부하로 보고 전류를 구하여 해를 구한다. 발전기 고정자 저항을  $r$ ,과도 리액턴스를  $L'_d$ , 발전기와 무한대 모선사이의 송전선로 저항  $R_o$ ,인덕턴스  $L_o$ 의 선로라 하고,  $R = R_o + r, X = X_o + X'_d$ 로 놓고 무한대 모선을 실효치 전압  $v_\infty$ 의 3상 평형이라 하면

$$v_{abc} = \sqrt{2} v_\infty \begin{bmatrix} \cos(\omega_R t + \alpha) \\ \cos(\omega_R t + \alpha - 120^\circ) \\ \cos(\omega_R t + \alpha + 120^\circ) \end{bmatrix}$$

이고 Park의 변환에 의해 전류를 구할 수 있다.

$$I_d = \frac{1}{R^2 + X^2} [ R(V_\infty \sin(\delta - \alpha) + E'_d) + X(V_\infty \cos(\delta - \alpha) - E'_q) ] \quad (11)$$

$$I_q = \frac{-1}{R^2 + X^2} [ -X(V_\infty \sin(\delta - \alpha) + E'_d) + R(V_\infty \cos(\delta - \alpha) - E'_q) ] \quad (12)$$

#### 5.2 블럭선도와 ODCS 표현.

ODCS 을 이용하여 시뮬레이션을 하기 위해서는 시스템을 묘사하는 각 방정식을 객체 단위로 표현해야한다. 블럭선도를 이용하면 이들 잘 표현할 수 있고 ODCS 로 쉽게 옮겨진다. 각

ODCS 표현을 연결하면 하나의 발전기 객체가 완성된다. 즉 기본적인 객체 부품을 조립하면 발전기 객체가 완성된다.

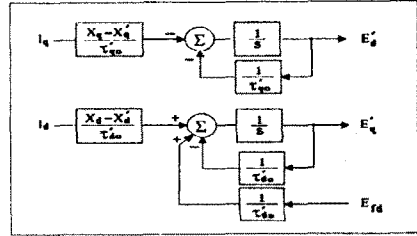


그림 4 과도 모델의 블럭선도

```

Integrator s1,s2;
Gain gain1,gain2,gain3,gain4,gain5;
Summation sum1,sum2;
s1.SetInit(Ed0,TimeStep);
s2.SetInit(Fq0,TimeStep);
gain1.SetGain((xq-x'q)/tau_d);
gain2.SetGain(1/tau_d);
gain3.SetGain((xd-x'd)/tau_d);
gain4.SetGain(1/tau_d);
gain5.SetGain(1/tau_d);
g1=gain1.gain(Iq);
g2=gain2.gain(Ed);
Sum1=sum1.sum(2,-g1,-g2);
E'd=s1.Integral(Sum1);
g3=gain3.gain(Iq);
g4=gain4.gain(E'q);
g5=gain5.gain(Efd);
Sum2=sum2.sum(3,g3,-g4,g5);
E'q=s2.Integral(Sum2);

```

그림 5 과도 모델의 ODCS 표현

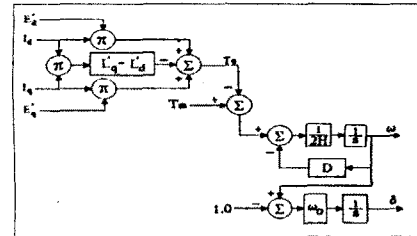


그림 6 토크와 동요 방정식의 블럭선도

```

Integrator s3,s4;
Summation sum3,sum4,sum5,sum6;
Multiply mul1,mul2,mul3;
Gain gain6,gain7,gain8,gain9;
s3.SetInit(omega_0,TimeStep);
s4.SetInit(delta_0,TimeStep);
gain6.SetGain(L'q - L'd);
gain7.SetGain(1/2H);
gain8.SetGain(D);
gain9.SetGain(omega_0);
m1=mul1.mul(2,E'd,Iq);
m2=mul2.mul(2,Iq,E'q);
m3=mul3.mul(2,Iq,Iq);
g6=gain6.gain(m3);
Te=sum3.sum(3,m1,-g6,m2);
Sum4=sum4.sum(2,-Te,Tm);
g8=gain8.gain(omega);
Sum5=sum5.sum(2,-g8,Sum4);
omega=s3.Integral(Sum5);
Sum6=sum6.sum(2,omega-1);
g9=gain9.gain(Sum6);
delta=s4.Integral(g9);

```

그림 7 토크와 동요 방정식의 ODCS 표현

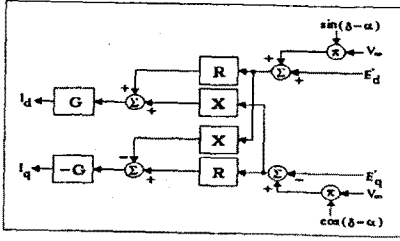


그림 8 전류 방정식의 블록선도

```

Multiply mul4,mul5;
Summation sum7,sum8,sum9,sum10;
Gain gain10,gain11,gain12;
Gain gain13,gain14,gain15;
gain10.SetGain(R);
gain11.SetGain(X);
gain12.SetGain(X);
gain13.SetGain(R);
gain14.SetGain(G);
gain15.SetGain(-G);
Vm1=mul4.mul(2, V_∞, sin(δ-α));
Vm2=mul5.mul(2, V_∞, cos(δ-α));
S1=sum7.sum(2,Vm1, E_d');
S2=sum8.sum(2,Vm2, E_d');
g10=gain10.gain(S1);
g11=gain11.gain(S2);
g12=gain12.gain(S1);
g13=gain13.gain(S2);
S3=sum9.sum(2,g10,g11);
S4=sum10.sum(2,g12,g13);
Id=gain14.gain(S3);
Iq=gain15.gain(S4);
  
```

그림 9 전류 방정식의 ODCS 표현

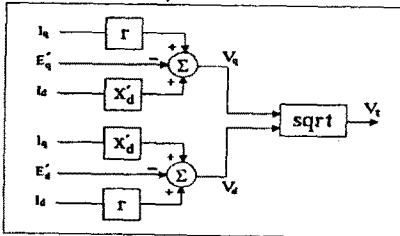


그림 10 단자전압 방정식의 블록선도

```

Gain gain16,gain17,gain18,gain19;
Summation sum11,sum12;
gain16.SetGain(r);
gain17.SetGain(x'_d);
gain18.SetGain(x'_d);
gain19.SetGain(r);
g16=gain16.gain(Iq);
g17=gain17.gain(Id);
Vq=sum11.sum(3,-g16, E'_q,g17);
g18=gain18.gain(Iq);
g19=gain19.gain(Id);
Vd=sum12.sum(3,-g18, E'_d-g19);
Vt=sqrt(Vd,Vq);
  
```

그림 11 단자전압 방정식의 ODCS 표현

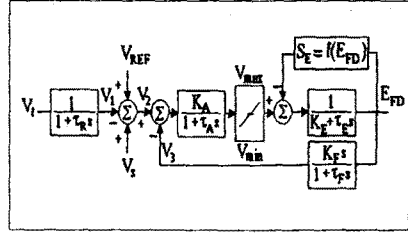


그림 12 IEEE type 1 여자시스템

```

Transfer tr1,tr2,tr3,tr4;
Summation sum11,sum12;
tr1.SetInit(τ_R,1,0,1,TimeStep);
tr2.SetInit(τ_A,1,0,K_A,TimeStep);
tr3.SetInit(τ_E,K_E,0,1,TimeStep);
tr4.SetInit(τ_F,1,K_F,0,TimeStep);
V1=tr1.trans(Vt);
V2=sum11.sum(3,V_REF,V1,Vs);
S12=sum12.sum(2,V2,-V3);
S13=tr2.trans(S12);
E_FD=tr3.trans(S13);
V3=tr4.trans(E_FD);
  
```

그림 13 IEEE type 1 여자시스템의 ODCS 표현

여자시스템의 포하는 무시하고 그림 8에서  $G = \frac{1}{R^2 + X^2}$  이다.

### 5.3 시뮬레이션

1기 무한대 모선의 시뮬레이션은 그림 5,7,9,11,13의 ODCS의 표현을 서로 연결함으로써 가능하다. 각 블록선도의 화살표 방향으로 단계적으로 계산이 이루어지고 같은 시간 간격으로 원하는 시간까지 반복계산한다. 시뮬레이션에 사용한 발전기의 역률은 0.85이고 선로는  $R_s=0.02$  pu,  $X_s=0.4$  pu, 무한대 모선의 전압은 1pu 이다. 발전기 데이터는 표 1 에 여자기 데이터는 표 2 에 나타났다.

표 1 발전기 데이터

구분	파라미터	수치(pu)
발전기	$x_d$	1.7
	$x'_d$	0.245
	$x_q$	1.64
	$x'_q$	0.38
	$\tau'_{do}$	5.9s
	$\tau'_{\infty}$	0.075 s
	$H$	2.37 s

표 2 여자기 데이터

구분	파라미터	수치(pu)
여자기	$\tau_R$	0
	$K_R$	1.0
	$\tau_A$	0.05
	$K_A$	400
	$\tau_E$	0.95
	$K_E$	-0.17
	$\tau_F$	1.0
	$K_F$	0.4

먼저 발전기 데이터를 이용하여 발전기 초기값을 계산한다. 이 초기값으로 각 객체를 초기화한다. 각 객체를 연결함으로써 시뮬레이션이 된다. 정상상태에서 5초가 지난후에 기계적 토크를 10 퍼센트 증가시키고 다시 25초가 될때 단자전압을 5퍼센트 증가시켜 시뮬

레이션을 했다. 발전기의 초기값은 표 3 과 같다.

표 3 발전기 초기값

발전기 초기값	$i_d$	-1.596
	$i_q$	0.701
	$i_F$	2.826
	$E_{FD}$	2.529
	$v_d$	-1.148
	$v_q$	1.675
	$T_e = T_m$	1.001

5.4 시뮬레이션 결과

시뮬레이션 결과는 그림 14에 나타났다. 정상상태에서 임의로 Tm 을 변화시킬때 단자전압의 변화는 미미하고 위상각은 새로운 동작 점으로 바뀌어 안정하게된다. 전류와 각주파수도 안정하게 됨을 알 수 있다. 기준 전압을 변화시는 단자전압은 이 변화를 따라가고 다른 값들은 진동이 그리 크지 않음을 알 수 있다.

그림 14의 결과는 분계쿠타의 해석 결과와 일치함을 확인했다. ODCS와 분계쿠타의 결과의 상대오차는 무시할 수 있는 정도이다.

6. 결론

동기발전기를 ODCS을 이용하여 모델링하고 1기 무한대 계통에 적용하여 시뮬레이션을 행하였고 그결과를 보였다. 동기발전기 기술 묘사하는 각 방정식을 블록단위로 구성하고 객체화하여 이들을 상호 연결함으로써 시뮬레이션이 가능하고 하나의 발전기 객체가 완성된다. 실제 계통은 다수의 발전기로 구성되어있고 이를 기존의 방법으로 시뮬레이션하기 위해서는 프로그램이 복잡하고 어렵다. 하지만 이미 구성된 발전기 객체를 이용한다면 프로그램은 간단하고 적은 노력으로도 시뮬레이션이 가능하다.

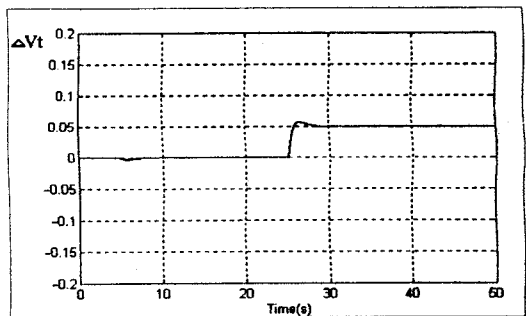
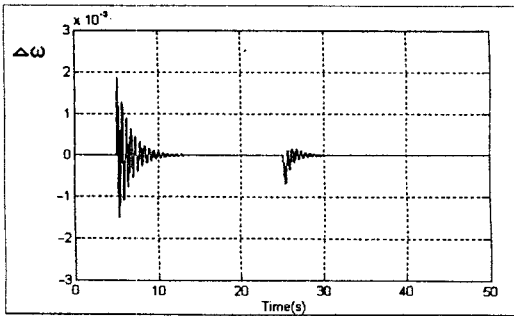
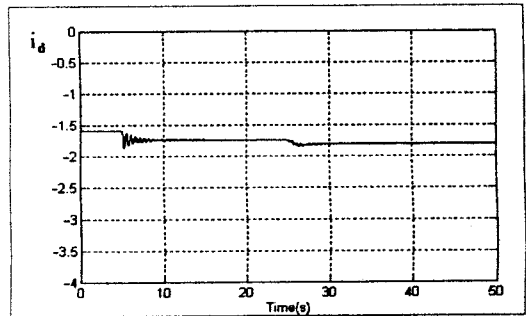
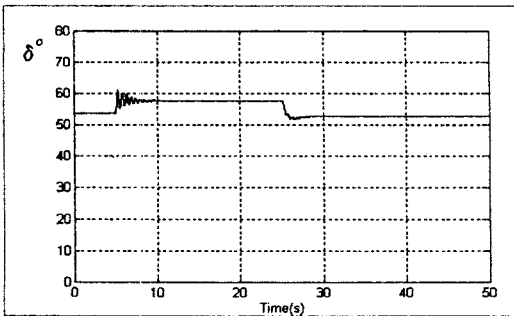


그림 14 기계적 토크의 10% 변화와 기준전압의 5% 변화에 대한 발전기 응답

7. 참고 문헌

- [1] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions," IEEE Transactions on Power Apparatus and Systems, Vol PAS-91, pp. 1643-1650, July/August 1972.
- [2] F. L. Alvarado, R. H. Lasseter, and J. J. Sanchez, "Testing of trapezoidal integration with damping for solution of power transient problems," IEEE Trans. Power Apparatus and Systems, Vol PAS-102, pp. 3783-3790, December 1983.
- [3] P.M. Anderson and A. A. Fouad, *Power System Control and Stability*, Iowa State University Press, 1977. pp. 254-263, pp. 400-412.
- [4] P.Kundur, *Power System Stability and control*, McGraw-Hill, 1994
- [5] M. L. James, G. M. Smith, and J. C. Wolford, *Analog Computer Simulation of Engineering Systems*, International Textbook company, 1967.
- [6] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, Addison-Wesley Publishing Company 1989.
- [7] 양해술, 최형진, *C++ 와 객체지향 프로그래밍*, 도서출판 정일, 서울, 1993.