AUTOMATIC GENERATION CONTROL
USING DIGITAL TECHNIQUES

BY

A. Abdul-Magid Hassan , Reda M. K. El-Dewieny
and Elsayed H. El-Konyaly

ABSTRACT:

   Techniques for simulation and application of automatic
digital generation control are presented. Objectives of
generation control are reviewed. A digital control scheme
is described and implemented. The presented directive soft-
ware is capable of guiding the whole system so that control
objectives are achived in a satisfactory manner.

1. INTRODUCTION:

   Over the past decade digital computers have been insta-
lled by many electricity supply authorities and utilities to
aid the control engineers and operators in both the operat-
ional planning phase and real-time control of power systems.
By far, the majority of computer installations to-date serve
in supervisory level; carrying out such duties as data logg-
ing, monitoring variables and checking for alarm conditions,
contingency evaluations and economic dispatch calculations[1].

   Recent advances in digital technology have resulted in
tremendous reductions in digital component costs and improve-
ments in reliability. These advances led to the use of
digital components as integral parts in industrial systems
and direct computer control has been the usual results of
this integration.

   Automatic generation control has evolved --
the time when the junction was performe-
the days of simple analog syste--
of sophisticated direct d'

   Generation control i
objectives[2,3]:

1) Matching area generatio
2) Distributing these chan,
   minimize operating costs
   constraints such as high
   derations.

_____

The authors are lecturers at
Engineering, Faculty of Engi

The use of digital computers and high speed data
communication systems allows a new degree of freedom per-
mitting, through software, the implementation of wide range
of control techniques.

The key to the derivation of improved control techniques
lies in the ability to test them on simulation models of the
process and controls.

This paper presents a digital control scheme applied
to a simulation model of a two-generating unit power-station.
Test and operating software are included.

2. SIMULATION MODEL OF A GENERATING UNIT:

A generating unit is fully simulated by a separately-
excited D-C motor drives synchronous generator. The output
torque of the D-C motor is either armature-or field-control-
led. The fastresponse is a cridet for the former control
scheme.

3. PROPOSED DIGITAL GENERATION-CONTROL SCHEME:

Fig. 1 shows the proposed digital control scheme in
which digital-to-analog converter (D/A), Analog-to-digital
conveter (A/D), multiplexers (selector) and data transmission
circuits form the necessary interface-requirements linking
the computer to the power system under control.

The actual most economic load allocation on the various
generating units are stored in the computer memory as a
table. Control action to each individual unit is primarily
a function of the unit's control error defined as the dif-
ference between desired and actual generation. This permits
tuning of control action to the individual unit's dynamic
response characteristics and provides unit control without
overshoot since control action is in proportion to the error
of the quantity being controlled.

Reallocation of generation may be guided through a com-
·and signal proportional to the frequency. This command
gnal comes from a tachometer mounting on the rotating
ft of the generating unit. The ability to relocate gene-
·n on command, offers a complete automatic feedback
l control system which can be of particular value in
ic implementation of system security measures.

upervisory routine which directs the interaction
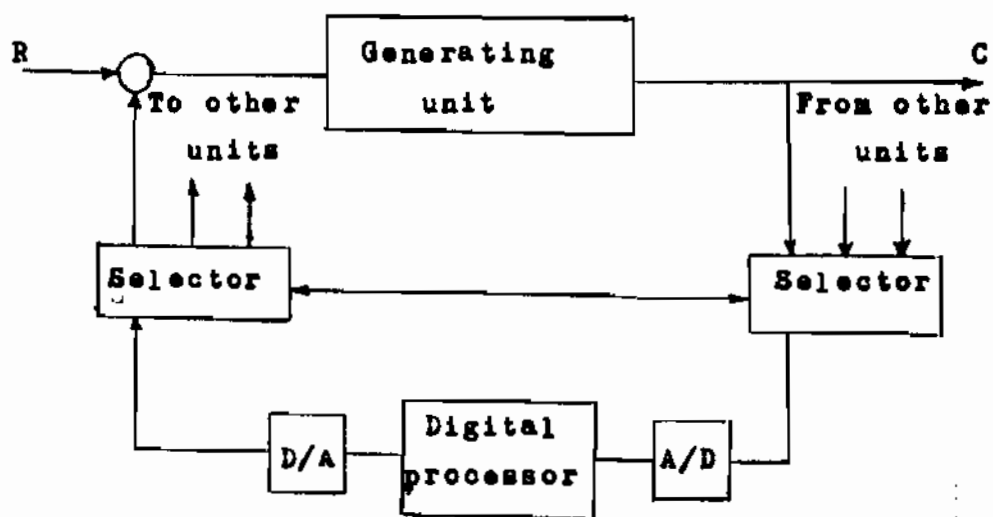ɔuter and controlled system is visualized by the
ɔ Fig. 2.

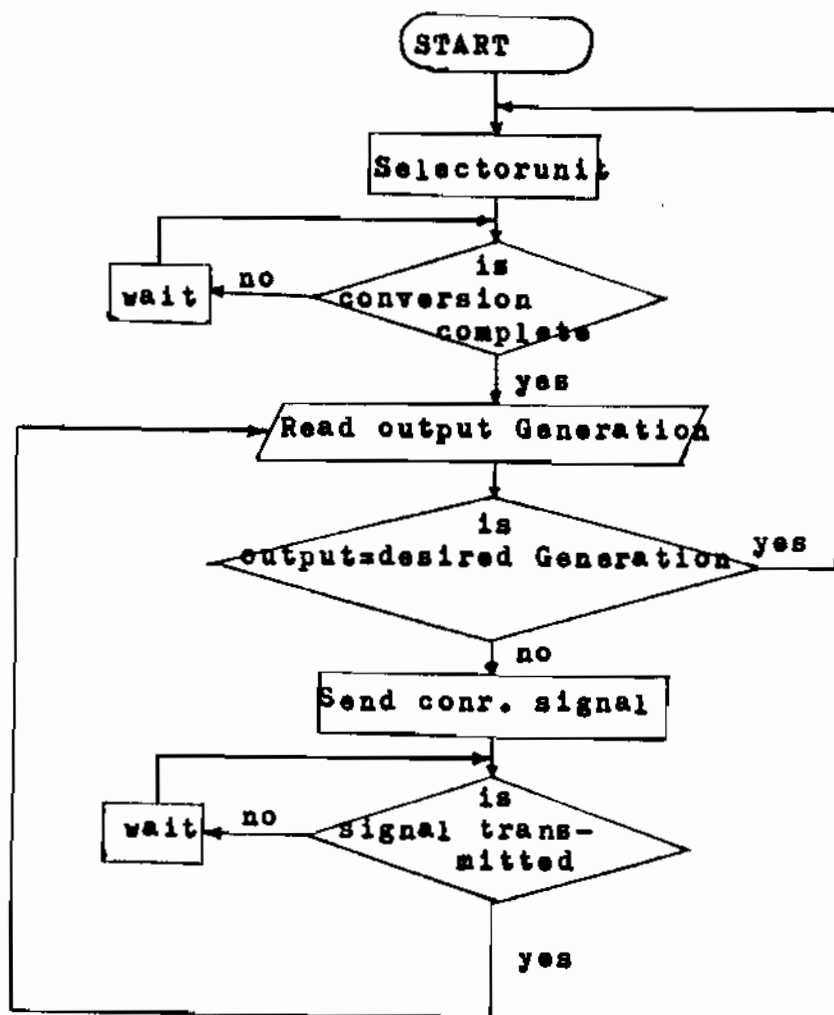Fig.(1): Digital Generation Control

Fig.(2): Flowchart of control logic.

Control actuation time depends on conversion speed
(or resolution) of the A/D used. Usually, such devices
have two extra bits (or pulses). Start conversion and
conversion complete are the two extra pulses which are
used as interrupt signals the time between which represe-
nts control actuation time.

4. SOFTWARE REQUIREMENTS:

Programming computers for direct applications and
control is usually carried out using Assembler Language
associated with the computer.

PAL-11 Assembler is the program which translates
symbolic language into absolute binary code capable of
being executed by the PDP-11 family of computers. The
symbolic language or source program in PAL-11A language
is composed of a sequence of statements which can be
edited and punched on a paper tape by the PDP-11 Text
Editor Program ED-11.

Accordingly, four extensive programs were developed
using the computer's assembly language and the text Editor.
Listing of the four programs is given in the appendix.
Program one tests and adjusts the performance of the
digital-to-analog converter. Program two examines the
closed loop embodying all devices of the digital control
scheme. Program three is a final program which directs
the control process on a two-interconnected power generat-
ing units.

If a forcasting routine is available to predict the
power demand in advance, it may be operated in conjuction
with program four which operates the system automatically
according to a time-power schedule.

According to program four the computer loads the whole
system automatically with a new load every two minutes which
is proportional to an octal number of 2100 in the program.

5. CONCLUSION:

Conclusively, automatic digital generation control is
implemented and is satisfactory working, using a simulation
model. Loading and control processes are programmed and
the necessary software is given. We are looking forward to
the day in which these simulation techniques are in practical
working order.

6. REFERENCES:

  1. K.J. Runtz et al., "Digital Control Scheme For a
          Generating Unit" IEEE-Trans., Vol. PAS,
          1973, pp. 478-483.

  2. F. P. DeMello et al., "Automatic Generation Control",
          IEEE-Trans., Vol. PAS-, 1973, pp. 710-722.

  3. H. J. Fiedler et al., "Automation Developments in the
          Control of Interconnected Electric Utility
          Systems", Proceeding of the IEEE, Vol. 57,
          No. 5, 1969, pp. 744 - 750.

7. APPENDIX:

    Software programs in assembly language.

7.1. Program One:

```
;TEST PROGRAM FOR THE DIGITAL/ANALOG CONVERTER

                AVEC = 310
              STAVEC = 312
              ADCOUT = 167754
               DACIN = 167752
               ADCSR = 167752
                KEYS = 177570
                  R0 = %0
                  R1 = %1
                  P2 = %2
                  R3 = %3
                  R4 = %4
                  R5 = %5
                   . = 3000
START:          MOV #340 , STAVEC       ;PRIORITY OF SEVEN DURING
                                        ;INTERRUPT
                MOV #4000 , AVEC        ;ADRESS OF SERVICE ROU-
                                        ;TINE TO INTERRUPT VECTOR
                MOV #100 , ADCSR        ;ENABLE INTERRUPT A
                HALT                    ;LOAD KEYS
BEGIN:          MOV KEYS , R5
                MOV R5 , KEYS
TEST:           MOV R5 , DACIN
                CLR R0
DELAY:          INC R0
                CMP R0 , #6250.
                BNE DELAY
                CMP KEYS , R5
                BEQ TEST
                JMP BEGIN
                 . = 4000
                RTI
                .END START
```

*

7.2. Program Two:

/L

```
;     TEST PROGRAM FOR THE CLOSED LOOP
;  CONTAINING THE A/D AND D/A CONVERTERS

AVEC = 310
STAVEC = 312
ADCOUT = 167754
DACIN = 167752
ADCSR = 167750
KEYS = 177572
R0 = %0
R1 = %1
R2 = %2
. = 1000
```

START:	MOV #342 , STAVEC
	MOV #4000 , AVEC
	MOV #100 , ADCSR
	MOV #101777 , R1
LOAD:	MOV R1 , DACIN
	CLR R2
LOOP:	INC R2
	CMP R2 , #6250.
	BNE LOOP
READ:	TSTB ADCSR
	BPL READ
	MOV ADCOUT , R0
	MOV R0 , KEYS
	DEC R1
	JMP LOAD
	. = 4000
	RTI
	.END START

*

7.3. Program Three:

```
; THE FINAL PROGRAM WHICH CONTROLS THE ECONOMIC POWER ALLOCATION
;         ON TWO INTERCONNECTED POWER UNITS

                    AVEC = 310
                   STAVEC = 312
                   ADCOUT = 167754
                    DACIN = 167752
                    ADCSR = 167750
                     KEYS = 177570
                       R0 = %0
                       R1 = %1
                       R2 = %2
                       R3 = %3
                       R4 = %4
                       R5 = %5
                        . = 3000
START:              MOV #340 , STAVEC        ;PRIORITY OF SEVEN DURING
                                             ;INTERRUPT
                    MOV #4000 , AVEC         ;ADRESS OF SERVICE ROU-
                                             ;TINE TO INTERRUPT VECTOR
                    MOV #100 , ADCSR         ;ENABLE INTERRUPT A
                    HALT                     ;LOAD KEYS
BEGIN:              MOV KEYS , R5
                    CMP R5 , #7777
                    BLE READY
                    MOV #7777 , R5
READY:             MOV #1000 , R4
LOOP:              CMP R5 , @R4
                    BLE STEADY
                    ADD #20 , R4
                    JMP LOOP
STEADY:            MOV R4 , R3
                    MOV @R3 , KEYS
LOAD2:             MOV R3 , R4
                    MOV #40000 , R2
                    ADD #4 , R4
                    ADD @R4 , R2
DONE2:             MOV R2 , DACIN
                    CLR R0
DELAY2:            INC R0
                    CMP R0 , #6250.
                    BNE DELAY2
                    MOV ADCOUT , R5
                    CMP R5 , #1153
                    BLE CONT2
                    MOV #41133 , R2
                    JMP ADJ2
CONT2:             MOV R3 , R4
                    ADD #54 , R4
                    CMP R5 , @R4
```

```
                        BLE LOAD1
                        SUB #30 , R4
                        MOV #40000 , R2
                        ADD @R4 , R2
ADJ2:                   MOV R2 , DACIN
                        CLR R0
WAIT2:                  INC R0
                        CMP R2 , #6250.
                        BNE WAIT2
LOAD1:                  MOV R3 , R4
                        ADD #12 , R4
                        MOV #100000 , R1
                        ADD @R4 , R1
DONE1:                  MOV R1 , DACIN
                        CLR R0
DELAY1:                 INC R0
                        CMP R0 , #6250.
                        BNE DELAY1
                        MOV ADCOUT , R5
                        CMP R5 , #2103
                        BLE CONT1
                        MOV #101000 , R1
                        JMP ADJ1
CONT1:                  MOV R3 , R4
                        ADD #46 , R4
                        CMP R5 , @R4
                        BLE TEST
                        SUB #14 , R4
                        MOV #100000 , R1
                        ADD @R4 , R1
ADJ1:                   MOV R1 , DACIN
                        CLR R0
WAIT1:                  INC R0
                        CMP R0 , #6250.
                        BNE WAIT1
TEST:                   CMP KEYS , @R3
                        BEQ SKIP
                        JMP BEGIN
SKIP:                   JMP LOAD2
                        .=4000
                        RTI
                        .END START
{
*
```

7.4. <u>Program Four</u>:

```
; THE FINAL PROGRAM WHICH CONTROLS THE ECONOMIC POWER ALLOCATION
;        ON TWO INTERCONNECTED POWER UNITS

                AVEC = 310
               STAVEC = 312
               ADCOUT = 167754
                DACIN = 167752
                ADCSR = 167750
                 KEYS = 177570
                   R0 = %0
                   R1 = %1
                   R2 = %2
                   R3 = %3
                   R4 = %4
                   R5 = %5
                 . = 3000
START:          MOV #340 , STAVEC       ;PRIORITY OF SEVEN DURING
                                        ;INTERRUPT
                MOV #4000 , AVEC        ;ADRESS OF SERVICE ROU-
                                        ;TINE TO INTERRUPT VECTOR
                MOV #100 , ADCSR        ;ENABLE INTERRUPT A
                HALT                    ;LOAD KEYS
INIAT:          MOV KEYS , R5
                MOV #1000 , R3
                CMP R5 , @R3
                BNE BEGIN
                MOV #41777 , DACIN
                CLR R0
STAY2:          INC R0
                CMP R0 , #6250.
                BNE STAY2
                MOV #101777 , DACIN
                CLR R0
STAY1:          INC R0
                CMP R0 , #6250.
                BNE STAY1
                JMP INIAT
BEGIN:          ADD #20 , R3
                CMP R3 , #2140
                BLE PAUSE
                MOV #1000 , R3
                JMP BEGIN
PAUSE:          CLR @#1016
                MOV @R3 , KEYS
LOAD2:          MOV R3 , R4
                MOV #40000 , R2
                ADD #4 , R4
                ADD @R4 , R2
DONE2:          MOV R2 , DACIN
                CLR R0
```

*

```
DELAY2:           INC R0
                  CMP R0 , #6250.
                  BNE DELAY2
                  MOV ADCOUT , R5
                  CMP R5 , #1153
                  BLE CONT2
                  MOV #41133 , R2
                  JMP ADJ2
CONT2:            MOV R3 , R4
                  ADD #54 , R4
                  CMP R5 , @R4
                  BLE LOAD1
                  SUB #30 , R4
                  MOV #40000 , R2
                  ADD @R4 , R2
ADJ2:             MOV R2 , DACIN
                  CLR R0
WAIT2:            INC R0
                  CMP R0 , #6250.
                  BNE WAIT2
LOAD1:            MOV R3 , R4
                  ADD #12 , R4
                  MOV #100000 , R1
                  ADD @R4 , R1
DONE1:            MOV R1 , DACIN
                  CLR R0
DELAY1:           INC R0
                  CMP R0 , #6250.
                  BNE DELAY1
                  MOV ADCOUT , R5
                  CMP R5 , #2103
                  BLE CONT1
                  MOV #101000 , R1
                  JMP ADJ1
CONT1:            MOV R3 , R4
                  ADD #46 , R4
                  CMP R5 , @R4
                  BLE TEST
                  SUB #14 , R4
                  MOV #100000 , R1
                  ADD @R4 , R1
ADJ1:             MOV R1 , DACIN
                  CLR R0
WAIT1:            INC R0
                  CMP R0 , #6250.
                  BNE WAIT1
TEST:             INC @#1016
                  CMP @#1016 , #2120
                  BNE SKIP
                  JMP BEGIN
SKIP:             JMP LOAD2
                  . = 4200
                  RTI
                  .END START
```