

# 위성 추적을 위한 HP-1000 Computer의 운용에 관련된 제반 연구

조중현, 고영철

1. 서론
2. HP-1000 Computer에서의 시각 출력과 Device Control
  - 2-1. 시각 출력 함수
  - 2-2. HP-IB상의 명령문과 변수 이동
3. 위성 추적용 기본 자료의 실제 계산
  - 3-1. 위성추적 가능 범위
  - 3-2. 정지 위성과 지구국간의 거리
4. 결론
5. 참고문헌

## 1. 서론

전파연구소에서 보유하고 있는 위성추적 시스템에서 안테나와 계측기의 운용 자동화를 위한 Computer로 HP-1000를 사용하고 있다.

이 Computer는 이러한 목적에 알맞게 매우 그 조작이 간단하고 Device Control하기 편리하게 제작되어 있다. 그러나 이 Computer가 보통 사용되는 범용과 그 O/S(Operating System)가 다르고 잘 알려져 있지 않기 때문에 운용법의 확실한 숙지가 필요하다. 이 보고서에서는 이 HP-1000 Computer 운용상 필수적이면서도 Technical Manual에서 소홀히 다루어 사용 곤란한 기능의 이용과 위성 추적을 할때 기본적으로 참고할 자료를 HP-1000 Computer를 사용하여 운용한 실례를 기술하였다. 또한 이 보고서는 Computer의 실행에 중점을 두어 계산식이나 자세한 내용은 기술을 피하였다.

## 2. HP-1000 Computer에서의 시각 출력과 Device Control

이 Computer는 원래 다른 기기의 원격 조종을 위한 목적으로 제작되어 있다. 따라서 다음과 같은 세가지 특징이 있다.

- 1) 실시간 실행(real time execution)
- 2) HP-IB채용(14개 기기까지 조종)
- 3) Device Table 채용(Fortran, Basic에서 직접 출력 가능)

이중 3)번항은 O/S가 처음 Boot-up될때 사용자가 조작하여 변경할 수 있다. 이 Computer의 O/S는 RTE-A로써 이미 밝힌 실시간 실행이 그 큰 목적이다. 처음 동력이 제공되어 Boot-up시 Batch File로 시각을 입력하여 내부 시계를 가동하고, High-Level언어(Fortran, Basic, Pascal)로 Program을 실행시킬때 특수 함수를 사용하여 시각을 출력해 내게 된다. 이러한 특수함

수들은 또 시간에 따라 연속적인 Program들의 실행도 가능하게 한다.

## 2.1 시각 출력 함수

HP-1000의 O/S인 RTE-A에는 Exec이란 함수를 Call할 수 있다. 마치 Subroutine을 호출하듯 사용할 수 있되 이것은 주로 주변기기와의 Data전송과 조종, 그리고 실시간 실행에 관계된 것들이다. 그러면 먼저 실시간 실행에 관계된 함수를 열거하면,

Exec 9, 10, 23, 24 : Program Scheduling

용 법;

(Call Exec(ecode, Name[, Pram \* 5[, bufr [, bufln]]))

ecode        9=대기 동작과 함께 즉시 실행  
               10=대기동작 없이 즉시 실행  
               23=대기 동작과 함께 Queue상태 유지  
               24=대기동작 없이 Queue상태 유지

name        Program 이름

Pram \* 5    5개까지의 Option 정수변수

bufr        Data를 주고받을 주소

Bufln       버퍼길이(+단어, -글자)

Exec 11 : 시각 호출

용 법;

(Call Exec(11, time[, year]))

Time        시각이 전송되어 오는 5단어 정수 배열

              Time(1)=1/100초 단위

Time(2)=1초 단위

Time(3)=분

Time(4)=시간(0-23)

Time(5)=해당연수초 부터의 날수(예 117)

Year        년도

Exec 12

용 법;

(Call Exec(12[, name[, units[, often[, hour[, min[, sec[, msec]]]]]]))

Units        시간 단위의 해상도 변환

0=시간 리스트에서 제거

1=1/100초        2=초

3=분                4=시간

Often        실행 되고자 하는 횟수

Hour        시작 시간(0-23, 기본=0)

Min        시작하는 분(기본=0)

Sec        시작한 초(기본=0)

mSec        시작하는 1/100초(기본=0)

이러한 특수함수중 특히 많이 사용되는 Exec 11 함수의 사용 예를 보면 다음과 같다.

List 1. 시간을 Computer로 부터 읽는 Program

```
Program time read
Integer time(5)
Call exec(11, time)
Print *, 'hour=', time(4)
Print *, 'min=', time(3)
Print *, 'sec=', time(2)
Stop
End
```

List 1의 Program을 실행 시키면 그 순간의 시각을 단말기 화면에 표시해 줄 것이다.

## 2.2 HP-IB 상의 명령문과 변수 이동

이 연구소에서 보유하고 있는 대부분의 계측기들은 HP-IB Bus를 채용하고 있어 그 운용을 자동화하기가 수월하다. HP-1000 Computer에는 현재 HP-IB 조종용회로를 1개 갖추고 있기 때문에 14개 까지의 기기를 조종할 수 있다.

HP-IB 상에서 기기조종용 명령어로는 주로 2-7개 문자로 이루어진 Mnemonic부호를 사용한다. 일례로 HP 8566B Spectrum Analyzer같은 경우 표1과 같이 따로 규정된 Mnemonic 부호를 이용하여 외부 Bus를 통한 명령을 받는다.

표 1. Hp 8566B Spectrum Analyzer의 주파수조종 외부 명령어

Mnemonic	수 행 기 능
CF	Center 주파수 지정
CS	Step 크기 합성
FA	최저 주파수 지정
FB	최고 주파수 지정
FS	기기에 내장된 최대 주파수폭 지정
KST	주파수대 고정
SS	Center 주파수 Step 크기 지정

이러한 명령어와 더불어 수치변수들이 사용되는 데 문자 변수나 수치변수 둘다 HP-IB상에선 ASCII 부호로만이 전송 가능하다. 문자변수같은 경우 Basic이나 Fortran등에서 쉽게 처리할 수 있어 Program으로 만들기가 쉬우나, 이러한 명령의 인수로써 주파수, Step, 기타 수치정보—같이 전송되어야 하는 수치변수는 Program내에서 계산이된 다음 ASCII 부호로 변환되어야 한다. Basic 같은 경우 수치변수와 문자변수의 변환이 매우 간결하나 Fortran의 경우에는 그 양상이 달라진다. 또 기기에 따라서는 16진수를 요구하기도 한다. List 2에는 이러한 대부분의 변환이 포함되는 Subroutine 을 소개한다. 이것은 Scientific Atlanta의 안테나 조종용 유니트에 Computer가 수치정보를, 전송할때 사용한 것이다.

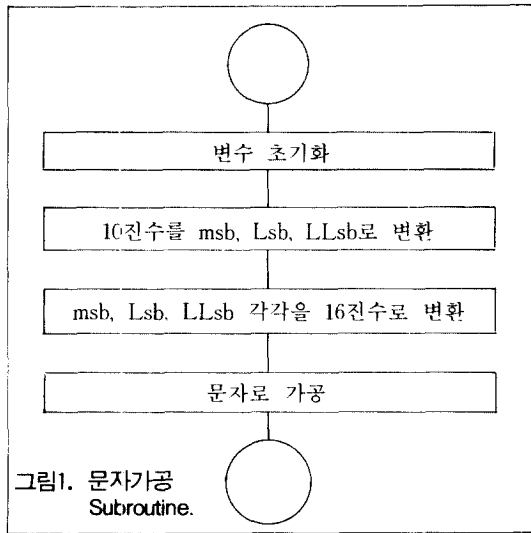
List 2. 10진수를 16진수 문자변수로 만드는 Subroutine

```

Subroutine decitohexa(degree, hexadeci)
real*8 degree
integer*4 msbdec, lsbdec, llbdec
character*16 hexa
character*6 hexadeci
character*2 msbstring, lsbstring, llbstring
character*1 fmsbst, rmsbst, flsbst, rlsbst, flldbst, rllbst
c initialization
msbdec=0
lsbdec=0
llbdec=0
c data formatting for acu internal data structure
do 10 ii=7,0:-1
  if((degree-360./2.**((8-ii).lt.0.))goto 10
  degree=degree-360./2.**((8-ii)
  msbdec=msbdec+2**((ii)
10 continue
  do 20 ii=7,0:-1
    if((degree-360./2.**((16-ii).lt.0.))goto 20
    degree=degree-360./2.**((16-ii)
    lsbdec=lsbdec+2**((ii)
20 continue
  do 30 ii=7,0:-1
    if((degree-360./2.**((24-ii).lt.0.))goto 30
    degree=degree-360./2.**((24-ii)
    llbdec=llbdec+2**((ii)
30 continue
c change the each significant bit data to hexa string
hexa='0123456789ABCDEF'
i=0
j=0
k=0
l=0
m=0
n=0
c msb manipulation
i=int(msbdec/16)+1
fmsbst=hexa(ii)
msbdec=msbdec-int(msbdec/16)*16
j=msbdec+1
rmsbst=hexa(j,j)
msbstring=fmsbst//rmsbst
c lsb manipulation
k=int(lsbdec/16)+1
flsbst=hexa(k,k)
lsbdec=lsbdec-int(lsbdec/16)*16
l=lsbdec+1
rlsbst=hexa(l,l)
lsbstring=flsbst//rlsbst
c llb manipulation
m=int(llbdec/16)+1
flldbst=hexa(m,m)
llbdec=llbdec-int(llbdec/16)*16
n=llbdec+1
rllbst=hexa(n,n)
llbstring=flldbst//rllbst
c total
hexadeci=llbstring//lsbstring//msbstring
return
end

```

List 2. 의 Flow Chart를 그려보면 그림 1과 같다.



### 3. 위성 추적용 기본 자료의 실제 계산

실제적으로 위성을 추적하다보면 각종 자료를 처리할때 필요한것 중에 지구국과 위성간의 거리와 극궤도위성의 추적한계등이 있다. 정지위성까지의 거리는 전계강도의 측정 전파특성을 알기위해 꼭 필요한 것이고, 추적한계는 시스템의 설계와 그 특성 검토시 꼭 필요한 것이다.

#### 3-1. 위성추적 가능 범위

위성은 지구와 위성의 무게 중심을 한 초점으로 하는 타원 궤도를 돈다. 이 위성이 상대적으로 낮은 궤도를 돌 경우 지구상의 한 지점에서 관측하면 그림 2에서와 같이 출몰 현상이 일어난다.

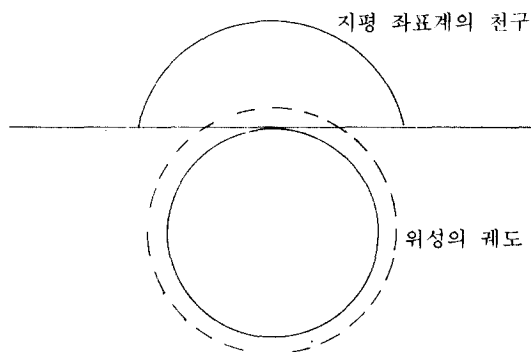
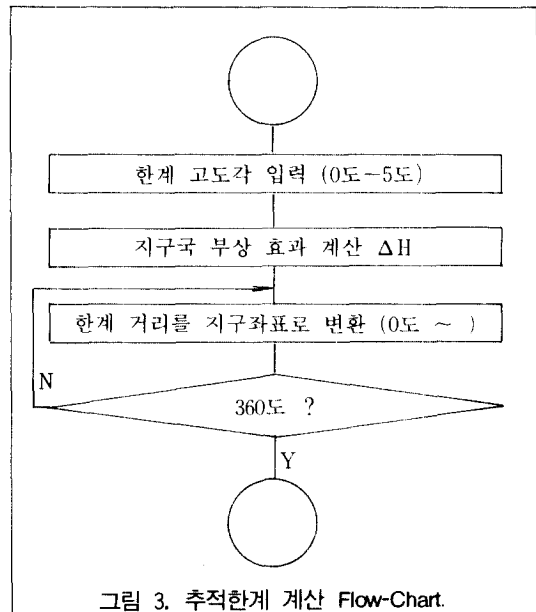


그림 2. 지구상의 한지점의 지평좌표와 위성 궤도의 출몰

이때 그 지점에서 지평좌표계를 그림 2와 같이 설정하면 우리는 그 위치에서 위성의 위치를 고도각과 방위각(A, h)으로 표현 할 수 있다. 이때의 지평면을 전 방향으로 1-5도 상승시키면 -Sky Line 때문에 -그 관측 우리는 이를 기하학적으로 계산하여 지구상의 한 지점에서 알려진 위성의 조건으로 이 위성을 추적할 수 있는 한계 범위를 구하였다. 이 Program의 Flow Chart 그림 3과 같고 완성된 Program은 List 3 과 같다.



List 3. 추적 한계 Program.

```

c evaluate the possible tracking area
c programed by J.H.Jo
  program tracango
  implicit real*8 (a-z)
  pi=3.141592653642
  radian=pi/180.
c latitude of site
  lat=37.388503*radian
c longitude of site
  lon=126.943277*radian
c elevation of sky line
  
```

```

c thetap=1.*radian
  write(1,20)
20 'format(5x, "input the elevation of the sky
  & line in Degree unit")
  read(1,*)thetap
  thetap=thetap*radian
c height of the satellite in km
  height=805.
c earth radius
  radius=6378.140
c raised height of site caused by the elevation of
c sky line
  x=(-2.*radius+dsqrt(4.*radius**2-4.*(1.+1./(dtan
  & (thetap))**2) & *(radius**2-(radius+height)**2)))/
  & (2.*(1.+1./(dtan(thetap)) & **2))
c available tracking range in radian
  yp=dacos((radius+x)/(radius+height))
  do 100 ww=0,360,10
  xp=ww*radian
c delta longitude of the limit of range in geographic
c map
  b=datan(-(dsin(xp)/(dcos(pi/2.-lat)*dcos(xp)-dsin
  & (pi/2.-lat) & /dtan(yp))))
c latitude of the limit of range in geograhic map
  phip= pi/2.-dcos(dcos(pi/2.-lat)*dcos(xp)-dsin
  & (pi/2.-lat) & *dsin(yp)*dcos(xp))
  longitude=lon+b
  longitude=longitude/radian
  phip=phi/radian
  write(6,39) ww, longitude, phip
39 format(5x,f4.0,5x,f8.4,5x,f8.4)
100 continue
  stop
end

```

### 3-2. 정지 위성과의 지구국간의 거리

지구국과 정지위성간의 거리를 구하는 방법은 2가지가 있다. 구면삼각법을 이용하는 것과 벡터좌표계를 이용하는 것이 그것이다. 그러나 여기서는 좌표계 이해를 돕기 위해 벡터좌표계를

사용하기로 한다. 이때 2가지의 가정이 필요하다.

- 1) 지구를 완전한 구로 가정한다.
- 2) 정지 위성은 적도 평면에 위치한다.

그림4는 기준좌표계이다.

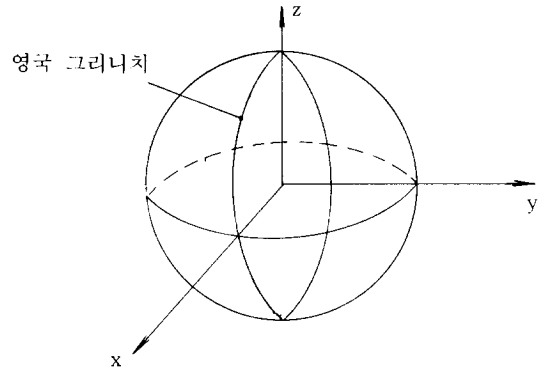


그림 4. 지구상의 기준 좌표계

먼저 지구중심으로 부터 위성 까지의 거리를 R 이라 하면 직교 좌표계에서의 각 성분에 대한 거리 벡터는 다음과 같다.

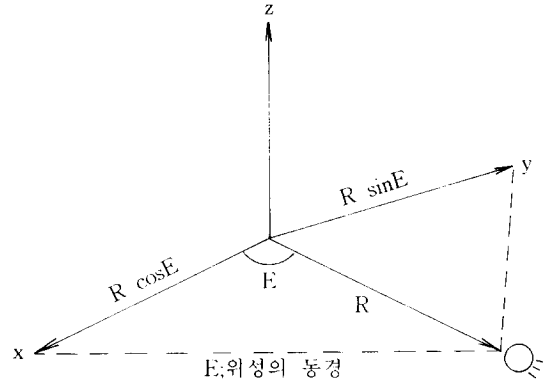


그림 5. 위성의 직교 좌표상의 위치.

이를 행렬로 표시하면 식 1과 같다.

$$\vec{R} = \begin{pmatrix} R \cos E \\ R \sin E \\ 0 \end{pmatrix} \quad (1)$$

또 우리 수신소의 위치를 직교 좌표계에서 표시하면 다음과 같다.

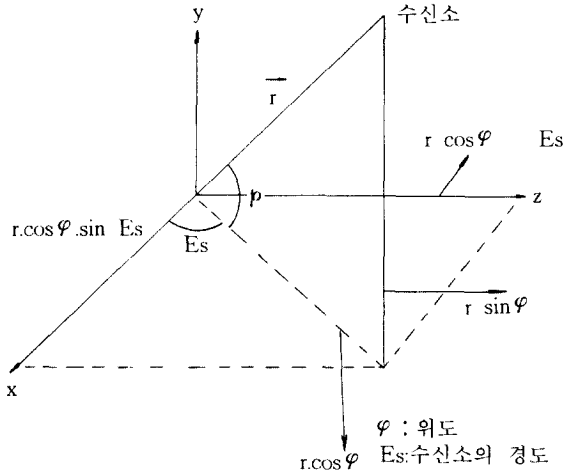


그림 6. 수신소의 위치 벡터

우리 수신소의 위치 벡터  $\vec{r}$ 의 스칼라량은 지구 반경으로 근사시킬 수 있다. 따라서 지구 반경을  $r$ 이라 가정하면  $\vec{r}$ 는,

$$\vec{r} = \begin{pmatrix} r \cos \varphi \cos E_s \\ r \cos \varphi \sin E_s \\ r \sin \varphi \end{pmatrix} \quad (2)$$

식 (2)와 같이 나타낼 수 있다.

따라서 두 위치간의 거리  $D$ 는 식(1)과 식(2)의 차로 나타낼 수 있다.

$$D = |R - r| = \sqrt{(R \cos E - r \cos \varphi \cos E_s)^2 + (R \sin E - r \cos \varphi \sin E_s)^2 + (r \sin \varphi)^2}$$

이렇게 구한 기본식을 응용하여 정지 위성에서 지구국간의 Path Loss를 계산하는 Program을 실행해 보았다. Path Loss를 구하는 공식은 이미 잘 알려져 있다. 여기서는 기준 주파수 1.7GHz에서 각 정지위성에 대한 Path Loss를 구하였다. 이의 Flow-Chart는 그림 7과 같고 Program은 List 4와 같다.

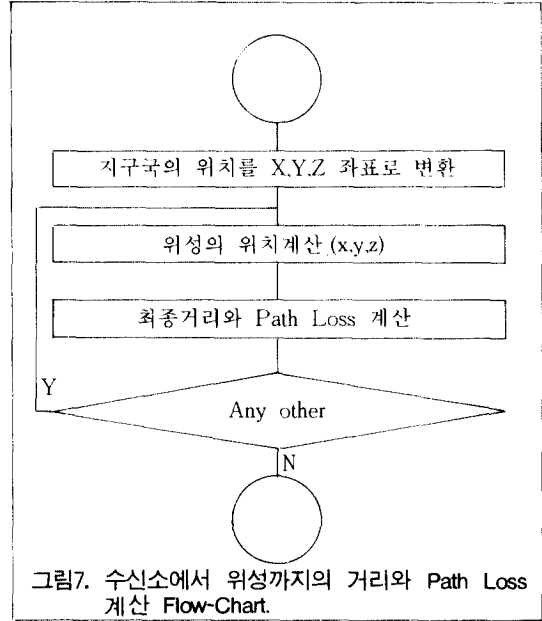


그림7. 수신소에서 위성까지의 거리와 Path Loss 계산 Flow-Chart.

List 4. Path Loss 계산 Program

```

program radiointensity
implicit real*8(a,z)
integer i
dimension longisat(10)
character*20 satname(10)
pi=3.141592653589
radian=pi/180.

c Input data unit=W.m
data longisat/63.179.66.174.140.136.110.124.0.0./
data satname/'INMARSAT(INTER-V)IND',
& 'INMARSAT(INTEL-V)PAC',
& 'INTELSAT-V IND', 'INTELSAT-V PAC',
& 'GMS-3', 'CS-2b',
& 'BS-2b', 'STW-1',
& 'NOAA maxim distance', 'NOAA minim distance'/

longiobs=126.94306
latitobs=37.3883
earthradi=6.378156d6
heightsat=3.6d7+earthradi
powersat=400

```

```

distance=4.d7
frequency=1.7d9
antenradi=5.
anteffec=.6
wavelength=3.d8/frequency
c calculate the distance between the satellite and station
stationX=earthradi*dcos(latitobs*radian)*dcos(longiob-
& s*radian)
stationY=earthradi*dcos(latitobs*radian)*dsin(longiob-
& s*radian)
stationZ=earthradi*dsin(latitobs*radian)
write(6,*) name longitude distance(m) pathloss &
& anten gain (db)
do 39 i=1,10
satelliX=heightsat*dcos(longisat(i)*radian)
satelliY=heightsat*dsin(longisat(i)*radian)
rho=dsqrt((stationX-satellix)**2+(stationY-
& satelliY)**2 & +stationZ**2)
if (i.eq.9)then
rho=3000000.
end if
if (i.eq.10)then
rho=1000000.
end if
c EIRP in W
dbW=10.*dlog10(powersat)
c Path Loss
Pathloss=20.*dlog10(4.pi*rho/wavelength)
c Antenna Gain
anteneffecarea=pi*antenradi**2*anteffec
antengain=
10.*dlog10(4.*pi/wavelength**2*anteneffecarea)
c total gain
totalgain=dbw-pathloss+antengain
Write(6,38) satname(i), longisat(i), rho, pathloss,

```

```

antengain. & totalgain
38 format(1x,a20,2x,f4.0,4x,f12.3,4x,f6.2,4x,f5.2,4x,f6.2)
39 continue
stop
end

```

## 4. 결 론

이상의 4가지 Program으로 HP-1000 Computer가 갖추고 있는 특수기능과 범용적인 측면을 살펴 보았다. 결과로는 HP-1000 Computer가 현재의 위성추적 시스템을 자동화하기에 충분한 기능과 용량을 갖고 있다는 것이 확인되었고, 실제 운용시 별다른 큰 문제를 수반하지 않았다. 그러나 몇가지 보완해야할 사항이 있는데 그것은 다음과 같다.

1) 언어 부족-Basic, Pascal 둘중 1개라도 보완되어야 한다.

2) UPS의 필요-가동시 순간 정전에 의한 자료 손실이 종종 있다.

3) 터미널 추가-많은 인원 사용시 필요하다.

이외에도 추후에 Image Processing 장비가 보강되면 두 시스템간의 연결에도 주의를 기울여야 한다. 또한 현재 이용이 곤란한 내부의 A/D Converter에 대한 S/W 적인 지원이 필요하다.

## 참고문헌

- Hewlett-Packard, 1985, HP 8566B Spectrum Analyzer Operating and Programming Manual.
- Hewlett-packard, 1984, HP-1000A, Computer Installation and Service.
- Pollack, S.V. 1983, Structured Fortran 77 Programming (Boyd & Fraser:Boston)