

SMART 해석코드 TASS/SMR 구동자 개편
Restructuring of the Driver for TASS/SMR, SMART Analysis
Code

이규형, 김수형, 황영동, 김희철

한국원자력연구소
대전시 유성구 덕진동 150

요 약

새로운 발전소인 SMART의 성능/안전해석을 위하여 기존의 TASS 코드에 일체형원자로의 고유한 모델을 추가하여 TASS/SMR 코드가 개발 중에 있다. 기존의 TASS에서는 변수참조를 위하여 외부 참조파일을 사용하고 있었으며 코드 관리와 변수 편집이 어려운 단점이 있었다. 변수의 동적 메모리 할당과 사용자 정의 데이터 타입이 가능한 Fortran90의 특성을 이용하여 TASS 구동자를 새롭게 작성하였다. 또한 기존의 대화식 입력과 utility 프로그램을 사용한 출력 파일 생성 대신 번호 입력을 사용하고 검증할 수 있도록 입출력 방식을 개편하였다. 코드의 검증을 위하여 개선된 구동자를 사용한 해석 결과와 기존의 구동자를 사용한 결과를 비교 확인하였으며 일치함을 확인하였다. 개편된 구동자의 사용은 향후 TASS/SMR 코드 개발과 사용자 입출력 친숙성을 용이하게 할 것으로 예상된다.

Abstract

TASS/SMR is under development by implementing the newly developed thermal hydraulic models for the integral reactor, SMART. The data structure of TASS utilizes external files, which could result in difficulties for the code maintenance and an easy understanding of the code. The driver for TASS/SMR was rewritten for the use of Fortran 90 advanced features, such as dynamic memory management and user defined derived type. The input structure of TASS changed from the interactive method to card numbered system. Also the output process of TASS reformed by removing the utility program. The validation of the TASS/SMR driver was performed by the comparison of results. The use of the revised TASS/SMR driver is expected to accelerate code development process as well as improve user friendliness.

1. 서론

TASS 코드는 CE형 및 Westinghouse형 원전의 인허가 및 재장전 과도기 사고해석에 사용된 열수력 해석코드이다. 현재 한국원자력연구소에서 개발하고 있는 SMART 일체형 원자로의 해석에 사용하기 위하여 일체형 고유의 모델을 추가하여 TASS/SMR 코드가 개발 중에 있다. TASS 코드는 FORTRAN77을 기반으로 작성된 코드이다. 따라서 새로이 개발된 FORTRAN90에서 제공하는 여러 가지 유용한 기능인 동적 메모리 할당, 사용자가 정의한 데이터 타입, 모듈의 정의 등을 사용하지 못하였다.

TASS 코드에서 사용하는 파일의 특성은 표 1에 표시되어 있다. 코드에서 각 부프로그램에서 사용하는 변수들 간의 통신을 위하여 외부 참조파일을 사용하고 있는데 외부 참조 파일의 형식은 표 2에 기술되어 있다. 외부 참조파일의 사용은 코드의 개발/변경 시 코드관리에 어려움이 있으며 변수의 배열 내 위치가 고정되어 변수의 크기변경, 변수의 추가 삭제가 어렵다.

TASS 코드에서 사용하는 입력 방식은 console이나 파일을 사용한다. 입력 형식은 대화식 (interactive) 방식을 채택하고 있다. 이 방법은 코드 수행 중 변수의 값을 확인하거나 바꾸는데 편리한 대신 인허가 측면에서 검증에 어려움이 있다. 계산 결과를 출력하기 위하여 TASS는 별도의 utility 프로그램을 사용한다. TASS 코드의 실행 순서는 그림 1에 나와 있다.

TASS/SMR 코드는 Fortran90에서 제공하는 여러 가지 유용한 기능인 동적 메모리 할당, 사용자가 정의한 데이터 타입, 모듈의 정의 등을 사용하여 참조파일을 프로그램 내로 흡수하였다. 입력의 검증을 위하여 대화식 입력 대신에 번호 입력을 처리하도록 코드가 수정되었다. 개선된 구동자에서는 입력에서 사용자가 지정하는 변수의 거동을 입력한 시간 간격으로 보여주도록 하여 plot 파일과 major output 파일을 생성하도록 하였다.

2. 코드 변수 정의

2.1 TASS driver

TASS 코드에서 사용하는 모든 변수의 통신은 다음의 방법으로 이루어진다. 헤더파일 ("header.h")에서 특성에 따라 8개의 큰 크기를 갖는 일차원 배열이 선언되고 공유되어 있다. 상기 헤더파일이 각 부프로그램에서 include되게 되고 실제로 사용하는 변수는 각각의 부프로그램에서 정의되고 공유된 8개의 일차원 배열과 EQUIVALENCE 문을 통하여 값을 공유한다.

```

include 'header.h'
.....
REAL          CHGS_RCS_FLOW                ( 4)
EQUIVALENCE ( CHGS_RCS_FLOW                ,RCS (30289))
REAL          CHGS_RCS_ENTH
EQUIVALENCE ( CHGS_RCS_ENTH                ,RCS (30293))
.....

```

TASS 코드에서 사용하는 변수의 특성은 표 2에 나와 있는 변수특성 파일의 형태를 갖고 있다. 외부 참조파일인 “variable.dat”은 변수특성 파일을 utility를 통하여 direct access 형태로 각 변수의 특성을 저장하고 있다. “index.dat” 파일은 변수특성 파일에서 변수 명칭의 alphabetic order에 따라 정렬을 시키며 line number가 적혀있다. TASS 실행에서 변수의 입출력을 수행하는 부분에서는 먼저 변수의 이름이 입력되면 “index.dat” 파일 내에서 변수의 위치를 읽어 들인 후 “variable.dat” 파일의 record number에서 그 변수의 특성을 읽어 들인다. 이때 변수의 입력과 출력은 8 개의 큰 크기를 갖는 공유된 일차원 배열 번호, 그 배열에서의 위치, SI - British unit conversion 번호를 FORTRAN의 내장함수인 CHAR을 이용하여 length 6 인 character 형태로 변환시켜서 사용하게 된다. 즉 입/출력인 경우 CHGS_RCS_FLOW(3) 같은 경우 block no = 4, offset no = 30292, conversion no = 34를 CHAR 함수를 통하여 “?040R” 번호를 부여 받게 된다. 이 번호를 다시 FORTRAN 내장함수인 ICHAR을 통하여 block no =4, offset no = 30292를 복원시켜 공유되어 있는 일차원 배열인 RCS(30292) 값을 참조하여 통신이 이루어지게 된다. 이와 같은 입/출력은 사용자가 작성한 입력 파일을 읽어 들일 때와 TASS에서 제공하는 시간에 따르는 pending action을 수행할 때 사용된다. 과도상태 계산에서 사용되는 restart 파일의 작성과 읽기는 8 개의 큰 크기를 갖는 8 개의 일차원 배열을 순차적으로 지정된 파일에 쓰거나 읽으므로 수행된다. TASS 코드에서 사용하는 pending action은 다음과 같다.

SetRamp : SetRamp 명령어는 과도기 모의 시간 중에 변수나 Array 값을 모의시간의 일차원 선형함수로서 입력한다. SetRamp 명령어는 SetRamp 수행의 시작 시간과 끝나는 시간을 제어할 수 있고, 또한 변수의 값으로도 일차원 선형 변화를 조절할 수 있는 기능을 갖고 있다.

Store : Store 명령어는 모의 중인 과도기의 과도기 시간에 따른 Data 저장 File을 6개의 File까지 제거, 수정 그리고 만드는 기능을 수행한다.

SetCondition : SetCondition 명령어는 과도기 모의 중 지정된 조건에 도달 했을 때 수행될 과도기 명령을 정의하거나 목록을 나타내거나 혹은 명령을 제어하는 기능을 수행한다. 과도기 명령은 과도기 모의중 조건이 만족되었을 때 Command File에서 명령이 지시되고 수행될 때에 File에 저장된다. 과도기 모의 중 SetCondition 명령 수행조건은 두 변수의 비교나 한 변수 값의 비교 및 Logical 값의 조건에 의하여 표현될 수 있다.

2.2 TASS/SMR driver

기존의 방법에서 TASS 변수의 외부 참조파일에 의하여 정의된 내용은 변수 이름과 그 특성 (type, unit conversion number, 8 개의 큰 공유 일차원 배열에서의 위치) 이다. 새롭게 도입된 방법은 외부 참조파일을 사용하지 않고 코드 내에 pointer로 변수의 특성에 맞게 derived data type을 큰 array를 정의하고 각각의 변수를 target attribute로 선언하여 연결시키는 방법이다. 이때 노드에 대하여 적용되는 변수와 패스에 대하여 정의되는 변수의 array size를 parameter를 통하여 compile 단계에서 array size를 동적으로 할당할 수 있게 설정하였다. 상기 방식의 코드 적용은 다음과 같이 구현하였다.

```

module all
! 변수체 (derived type VAR_COM) 구조 선언문
!
type var_com
character (len=32) :: lname = ' '
character (len=2) :: var_type = ' '
integer :: dim = 0
integer :: d1 = 0
integer :: d2 = 0
integer :: d3 = 0
integer :: con_f = 0
end type var_com
!
! 변수체 (derived type VARI) 구조 선언문
!
!
type vari
type (var_com) var
real, pointer :: real_sgn => null ()
real, pointer :: real_d1(:) => null (), real_d2(:,) => null(), real_d3(:,) => null()
integer, pointer :: int_sgn => null ()
integer, pointer :: int_d1(:) => null (), int_d2(:,) => null(), int_d3(:,) => null()
logical, pointer :: lo_sgn => null ()
logical, pointer :: lo_d1(:) => null (), lo_d2(:,) => null(), lo_d3(:,) => null()
character (len=70), pointer :: chr_sgn => null ()
character (len=70), pointer :: chr_d1(:) => null ()
end type vari

type (vari) xxx (1500)
end module all

module common_data
!
! able to dynamically allocate the maximum node number simulated here
integer, parameter :: n_node = 100
! able to dynamically allocate the maximum path number (inculding momentum+non-momentum)
simulated here
integer, parameter :: n_path = 200
! able to dynamically allocate the maximum momentum path number simulated here integer, parameter
:: n_mom_path = 150

.....
.....
real, target :: NODE_AREA ( n_node )
real, target :: NODE_HEIGHT ( n_node )
real, target :: N_BOT ( n_node )
real, target :: N_HEAT_XFER_LIQ ( n_node )
.....
.....
end module common_data

subroutine var_init
use all
use common_data
.....
.....
xxx( 19)% var = var_com ("NODE_AREA", "RE", 1, n_node, 0, 0, 52)
xxx( 19)%real_d1 => NODE_AREA

```

```

xxx( 20)% var = var_com ("NODE_HEIGHT", "RE", 1, n_node, 0, 0, 51)
xxx( 20)%real_d1 => NODE_HEIGHT
xxx( 21)% var = var_com ("N_BOT", "RE", 1, n_node, 0, 0, 51)
xxx( 21)%real_d1 => N_BOT
xxx( 22)% var = var_com ("N_HEAT_XFER_LIQ", "RE", 1, n_node, 0, 0, 43)
xxx( 22)%real_d1 => N_HEAT_XFER_LIQ
.....
.....
end subroutine var_init

```

새롭게 변경된 변수정의 방법에서는 모든 변수가 module 내에서 정의가 되며 그 변수들을 사용하는 부프로그램에서 use 명령을 통하여 변수간 통신이 이루어지게 된다. 변경된 방식은 그림 2에 표시되어 있다.

변경된 방식을 사용하면 외부 참조파일의 사용이 필요 없어지며 모든 변수의 정의가 코드 내에서 이루어지므로 코드의 개발과정에서 코드 유지관리가 용이해지게 된다. 또한 동적 메모리 할당방식이 도입되므로 계산 대상 플랜트의 특성에 따라 노드 및 패스 관련 변수의 array 크기를 조절할 수 있다. 디버그 과정에서 이전 코드에서 사용하던 character (length = 6) 인 참조번호가 없어져서 직접 변수를 access 할 수 있다.

3. 코드 입/출력 방식

3.1 TASS

TASS 코드에서는 입력방식으로 사용자가 console이나 파일을 통하여 개별 변수 값에 대한 입력과 job control을 대화식 (interactive) 으로 처리하도록 하는 방식을 택하고 있다. 기존의 입력 예를 보면 다음과 같다.

```

.....
.....
CORE_HT_AREA=          821.34
CHT_NNOD_BO=           12
CHT_NNOD_FC=           12
CHT_NREGIONS_RAD=     5
.....
.....
TIME_SCALE=0.005
CTL_TIME_STEP = 0.005
GO 100.
SNAPSHOT (SMART_STEADY,replace) STATIC (SMART_STEADY,replace)
quit

```

출력방식으로는 시간 변화에 따르는 개별 변수의 변화를 pending action인 Store 명령을 통하여 fixed된 파일로 생성하고 과도계산 종료 시 utility 파일을 사용하여 미리 생성된 파일로부터 사용자가 이용할 수 있는 ASCII 형태의 파일을 생성하도록 하고 있다.

3.2 TASS/SMR

대부분의 열수력 해석 코드에서 사용하는 입력 방식은 입력 내용의 검증과 사용자 편

의성을 고려하여 카드 번호가 부여된 값을 일괄적으로 (batch) 읽어 처리하는 방식을 채용하고 있다. TASS 코드의 입력방식을 변경하여 모든 입력 (변수 값의 입력 및 job control)에 대하여 카드 번호를 부여하여 입력을 단순화 시킬 수 있도록 하였다. 이때 추가된 부프로그램은 4 개 (card_input.f90, card_input_check.f90, card_read.f90, card_read_control.f90) 이다.

```
! problem type      rest file name
100 stdy

! time step control
! start time      time_card_number
1000 0.           5

1001 10.          0.0005      1.0           10.
1002 60.          0.001        1.0           10.
1003 160.         0.005        1.0           10.
1004 260.         0.01         1.0           10.
1005 661.         0.01         1.0           10.                smart_steady

!plot control-----
2000 5

2001 CTL_CORE_POWER_FRACTION
2002 SMR_SG_AREA_FACTOR(1)
2003 PRESS(3)
2004 CHT_TEMP_COOL(1)
2005 RCS_CHT_FLOWIN
.....
! name_node          n_hetero  ne_candidate  non_eq_state
200101 Core          F          0             0
! node_area          node_height  n_bot         n_geom
200102 1.64220       2.40000     1.24600      0
! press              press_stm
200103 1.500000E+07  0.00000
! temp_liq           temp_stm
200104 563.00000     0.00000
! lev1_mix
200105 2.40000
! n_heat_cap         n_heat_xfer_liq  n_heat_xfer_stm
200106 1.928280E+07  36096.9     28978.5
.....
end
```

또한 입력 방식에서 제어 입력을 단순화 시켜 다음의 입력을 가능하도록 하였다

```
이전 TASS 코드 입력 예
DBADD= 202, 30288, 2031, 7*0, 10*0, 10*0, 4010, 9*0\
9*0, 75, 40*0, 8*0, 12546, 1326
DBCOM= 4, 4, 6, 7*0, 10*0, 10*0, 4, 9*0\
9*0, 6, 40*0, 8*0, 1, 6
ELEMS= 1, 1, 1, 1, 1, 1, 2, 2, 1\
3, 3, 1, 1, 1, 1, 186*0

개선된 TASS/SMR 코드 입력 예
```

400600	"T-control variable"				
400601	6	2	SUB	GROUT(1)	GROUT(5)
400602			DIV	1	3.0
400700	"N-control variable"				
400701	7	2	SUB	CTL_CORE_POWER_FRACTION	GROUT(4)
400702			MULT	1	50.0
400800	"T+N control variable"				
400801	8	1	ADD	GROUT(6)	GROUT(7)

추가로 프로그램 수행도중 원하는 변수의 거동과 주요 계통 변수의 출력 값을 ASCII 형태의 파일로 생성 할 수 있도록 하는 방식이 도입되었다. TASS/SMR 코드의 실행순서는 그림 3에 나와 있다.

3.3 코드의 검증

코드 구동자의 변경에 대한 검증으로 TASS 구동자를 사용한 코드 입력계산 결과와 TASS/SMR 구동자를 사용한 내용에 따르는 입력의 계산 결과를 비교하였으며, 같은 결과가 얻어짐을 확인하였다 (그림4, 5).

4. 결론

상용 발전소 안전 및 성능해석에 사용된 TASS 코드를 SMART 원자로의 안전 및 성능해석에 사용하기 위하여 코드 개발이 진행 중에 있다. TASS/SMR 코드 개발을 용이하게 하기 위하여 다음 변경이 구동자에 대하여 이루어 졌다.

- ① Fortran77 으로 작성된 TASS 코드를 Fortran90 format에 맞게 새로이 compile 하였다.
- ② 외부 참조파일을 통한 변수간의 통신 방식을 pointer, target을 사용한 방식으로 변경하였다.
- ③ 입력 형식을 기존의 대화식에서 카드 번호를 따르는 방식으로 변경하였다.
- ④ 출력에서 plot 파일 생성하는 방식을 외부 utility 파일의 사용을 제거하였다.

코드 변경의 검증은 이전 코드 입력과 개편된 내용에 따르는 입력의 출력을 비교하여 같은 결과가 얻어짐을 확인하였다. TASS 코드 구동자의 개편을 통하여 향후 TASS/SMR 코드의 개발 및 유지관리의 편의성이 향상 되었다.

5. 참고문헌

- [1] 심석구, "TASS 코드 특정 기술보고서", KAERI/TR-845-1/97, 한국원자력연구소, 1997.
- [2] 이규형, "TASS/SMR 입력 기술서", -발간예정.

표 1. TASS 코드 사용 파일 특성

파일 형식	file name	file ext.	unit no.	I/O	type
Backup	-	bkp	38	Read	backup
Dynamic data	-	dyn	37	Write/read	dynamic data
Static data	-	stc	36	Write/read	static data
Store	tass000	str	39	Write	list
Command	-	cmd	100	Read	command
SetCondition	tass000	cnd	55	Write	condition
Variable index	index	dat	35	Read	변수 색인
Variable dictionary	variable	dat	34	Read	변수 기술
Log	tass000	log	32	Write	log
Journal	tass000	jnl	33	Write	journal
Automatic command	autotass	cmd	31	Read	default command
Units conversion	convert	cnv	30	Read	unit conversion

표 2. 외부 참조파일 특성

LNAME	TYP	DIM	D1	D2	D3	BLK	OFF	DESCPT	UNIT	GNV
PLT_DATA	RE	1	15000			1	1	Restore-only plant data	block	0
PLT_CHT	RE	1	15			1	1	Core heat transfer model data constants	subblock	0
CORE_HT_AREA	RE	0				1	1	Core heat transfer area, clad/coolant	m ²	52
CHT_NNOD_BO	IN	0				1	2	No. of axl. nodes in p.b. mesh, max 12	Counts	0
CHT_NNOD_FC	IN	0				1	3	No of ax nods in fc mesh, divbl into _BO	Counts	0
CHT_NREGIONS_RAD	IN	0				1	4	Number of radial regions in h.t. mesh	Counts	0
CORE_HYD_DIA	RE	0				1	5	Core hydraulic diameter	m	51
CORE_RAD_OUT_DUMMY	RE	1	3			1	6	Dummy	m	51
FUEL_DENSITY	RE	0				1	9	Density of uranium fuel	kg/m ³	91
PLT_RCS	RE	1	7840			1	16	Restore-only RCS plant data	Subblock	0
PLT_RCS_DESIGN	RE	1	10			1	16	RCS design features	Subblock	0
NUM_SG	IN	0				1	16	Number of steam generators	Counts	0

NAME ; 변수 명칭 (32자 이내)

TYP ; 변수 타입 (RE: real, IN: integer, LO: logical, CH: character)

DIM ; dimensionality (0: scalar, 1: 1-dimensional array, 2: 2-dimensional array)

D1 ; size of 1st dimension

D2 ; size of 2nd dimension

D3 ; size of 3rd dimension

BLK ; number of common block

1 (plant data) = PLT (15000)

2 (core heat transfer) = CHT (1024)

3 (core power) = POW (1536)

4 (RCS global common) = RCS (50000)

5 (SG secondary global common) = SGS (1536)

6 (control, protective and BOP system) = CTL (2048)

7 (user global common) = USR (1024)

8 (three dimensional core) = COR (1000)

OFF ; offset from the beginning of the common block

DESCPT ; 변수 description

UNIT ; 변수 unit (SI)

CNV ; 변수 conversion reference (SI → British)

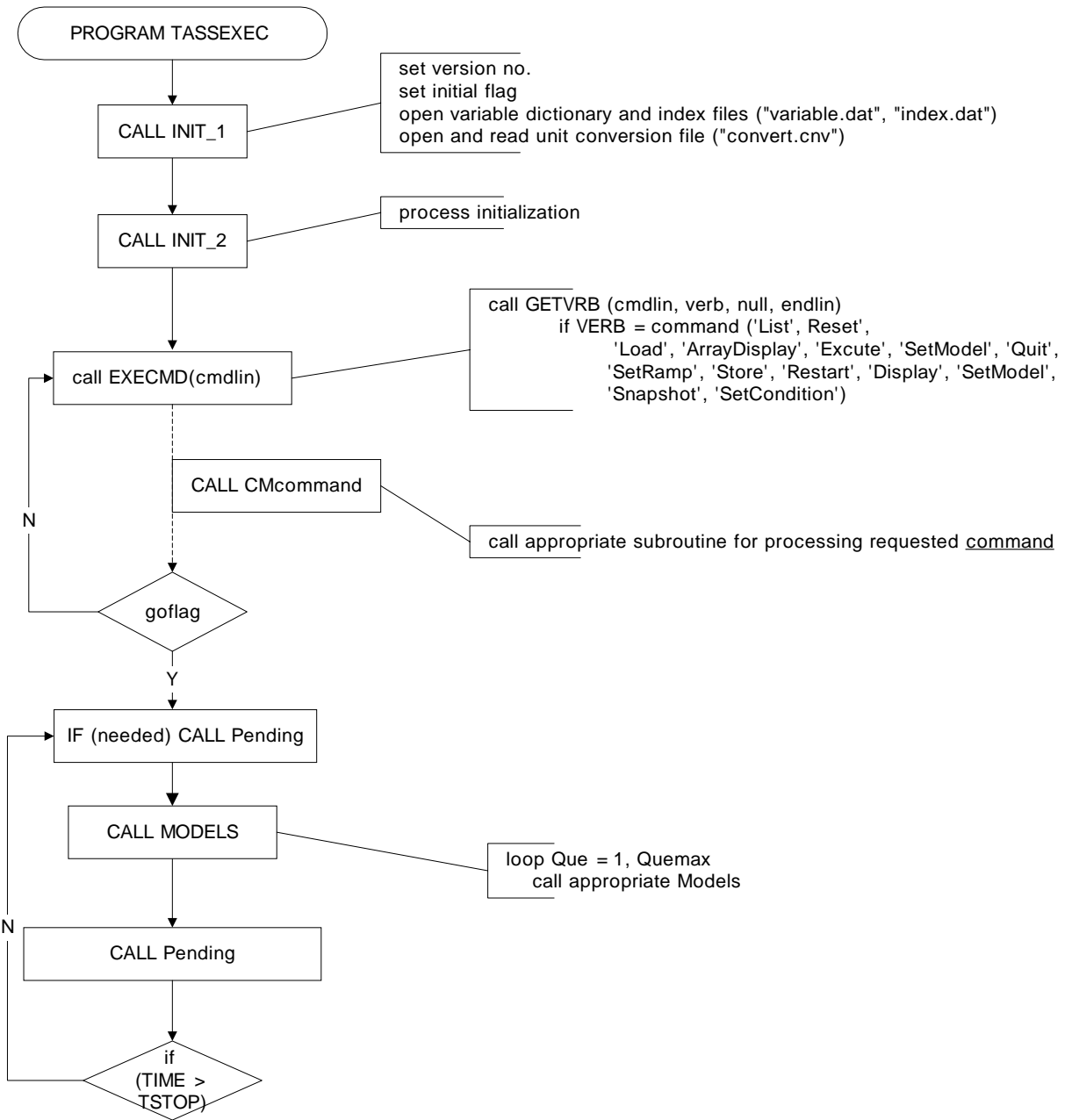


그림 1 TASS 코드 실행 순서

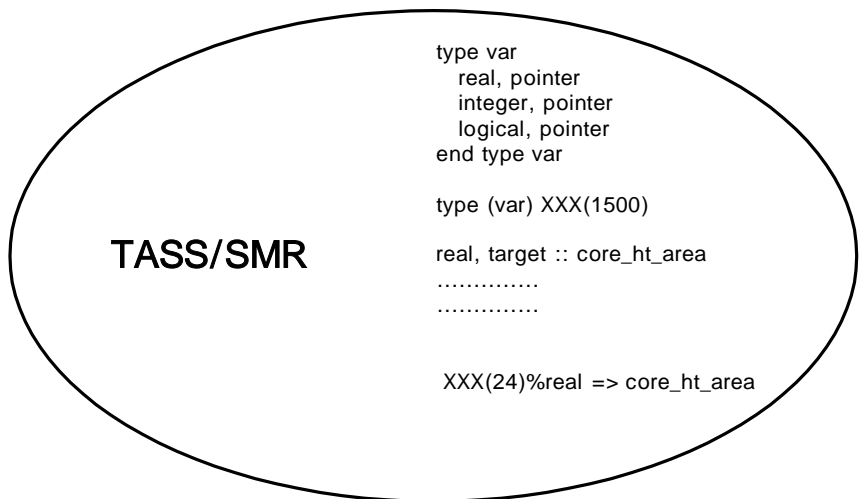
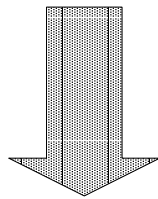
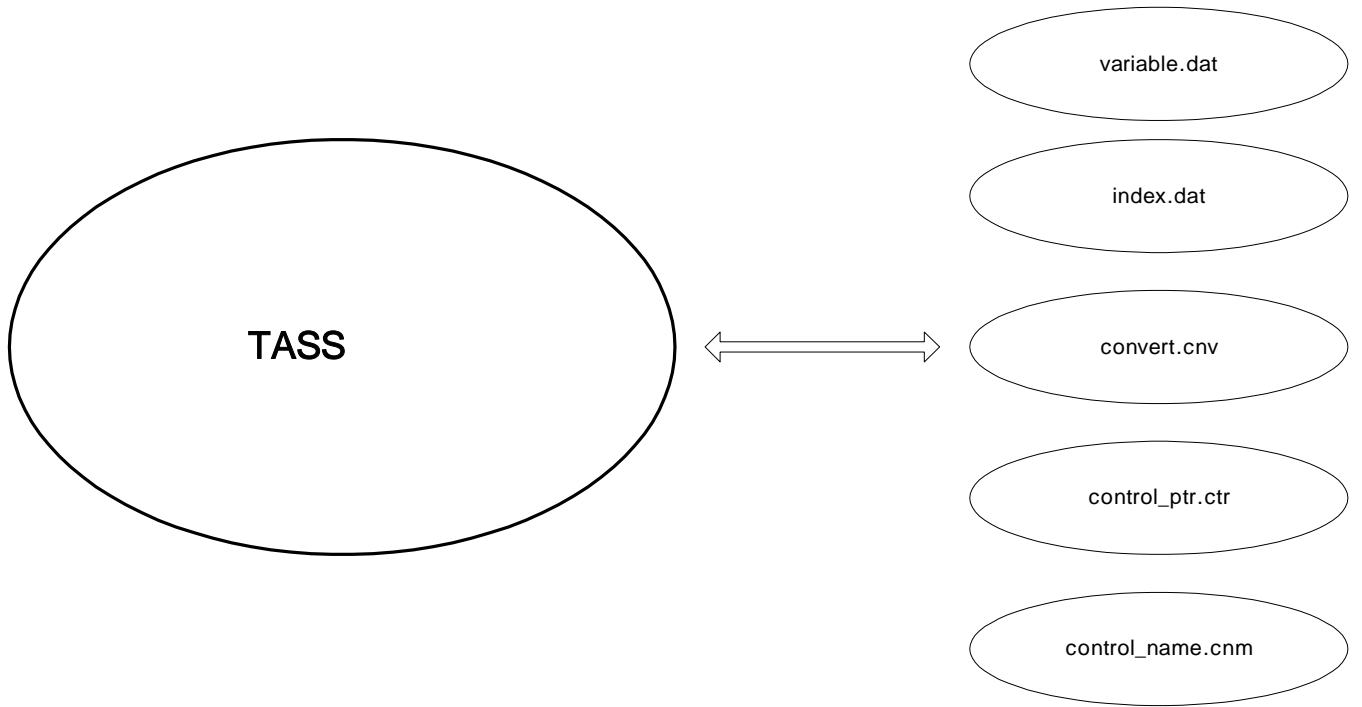


그림 2 코드 변경 내용

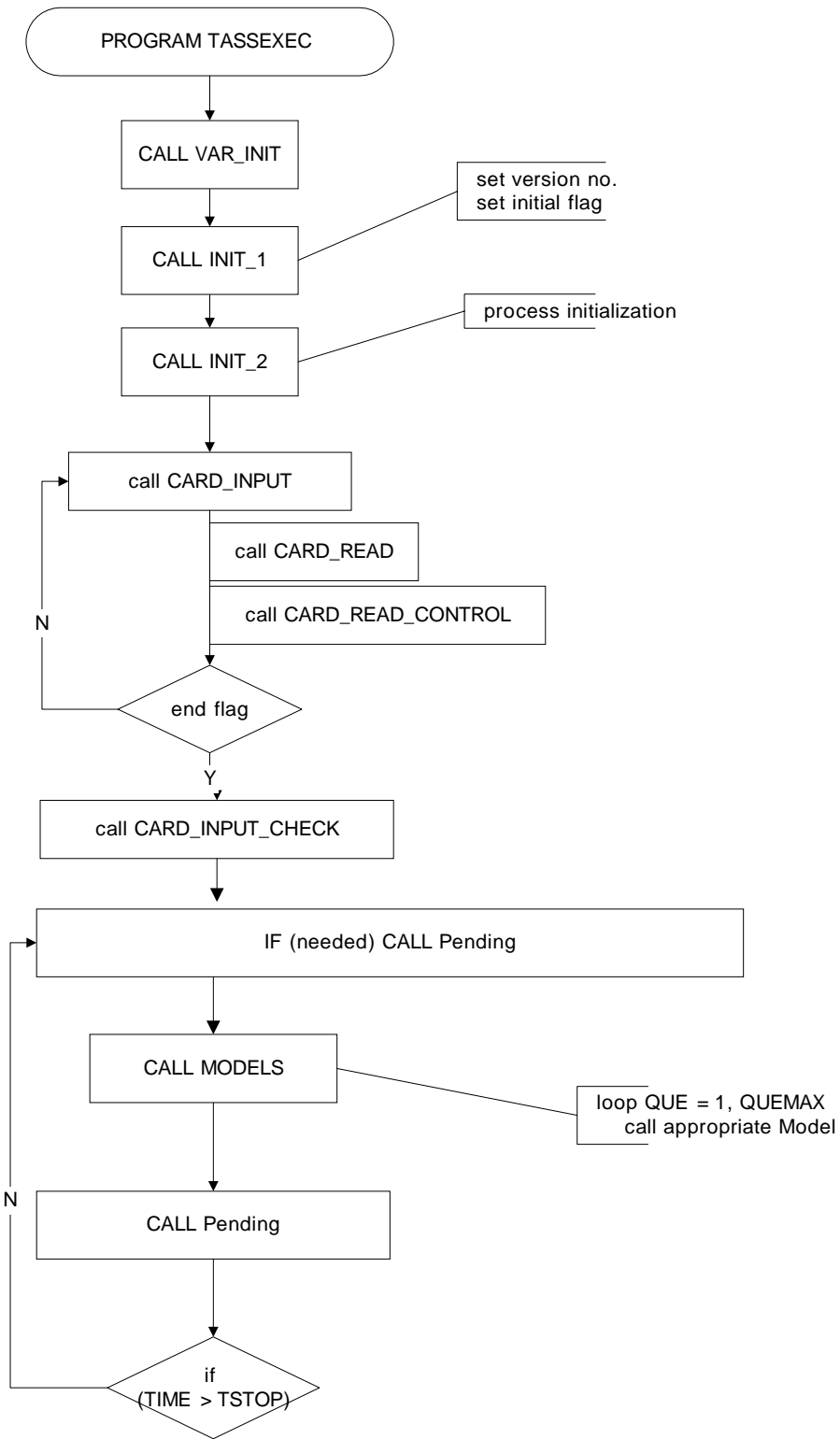


그림 3 TASS/SMR 코드 실행 순서

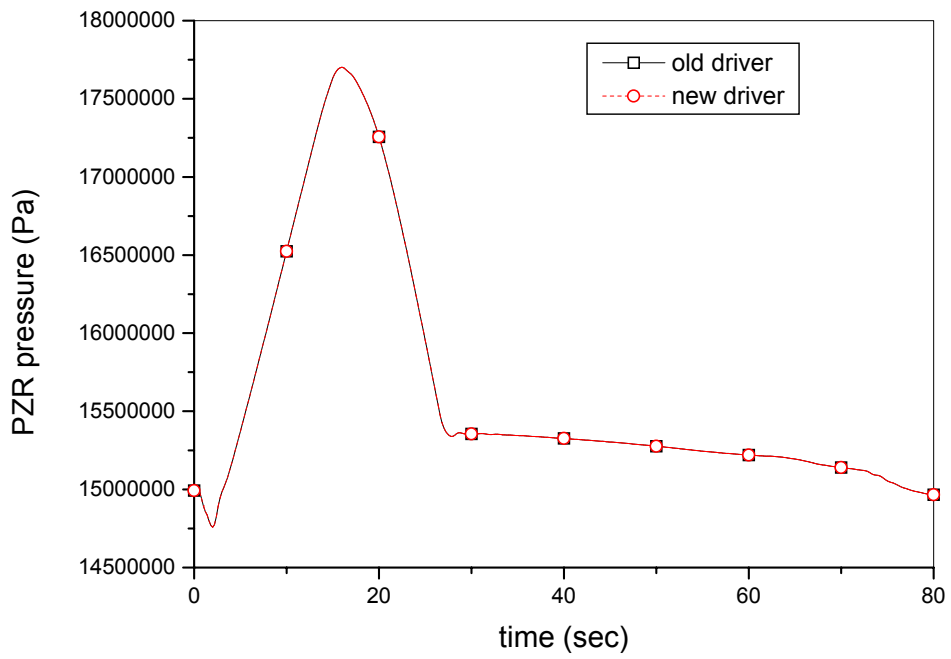


그림 4 SMART 급수관파단 사고에서 가압기 압력 변화

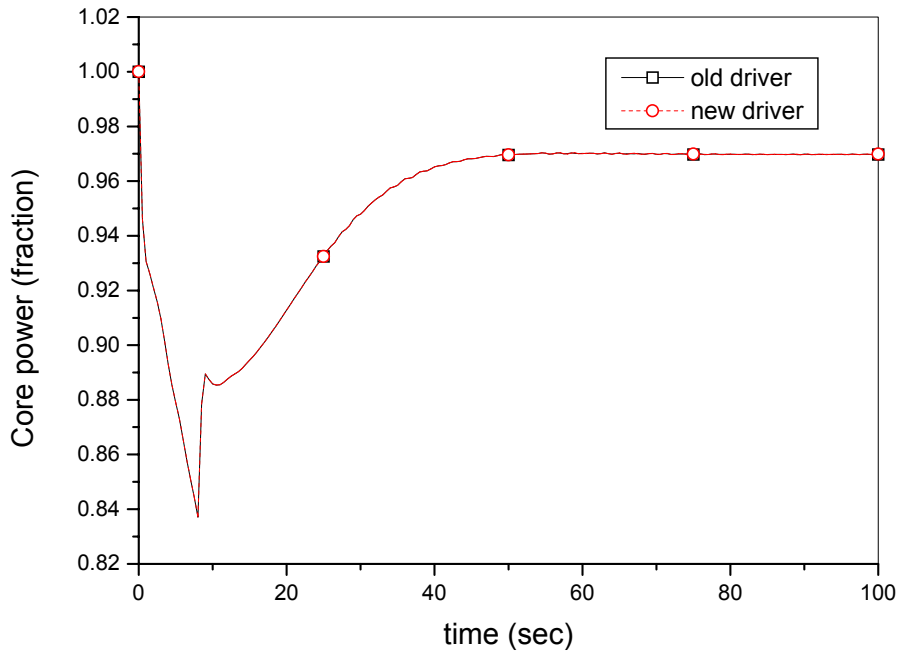


그림 5 SMART 제어봉 낙하사고에서 노심출력 변화