ANL/NE-16/19

The SAS4A/SASSYS-1 Safety Analysis Code System

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Document Availability

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (http://www.osti.gov/scitech/)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

ANL/NE-16/19

The SAS4A/SASSYS-1 Safety Analysis Code System

Chapter 6:

Control System

R. B. Vilim
Nuclear Engineering Division
Argonne National Laboratory

March 31, 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbol	Definition	Units
C	Normalized signal change	
D	Dynamic block operator	
e	Error estimate	
F	Static block operator	
i	Block index	
J	Block input signal	
j	Time step index	
K	Block type index	
k	Block type index	
L	Block output signal number	
m	Block output signal number	
q	Block input signal	
r	Block input signal	
S	Signal value	
t	Time	s
u	Input variable	
x	State variable	
y	Output variable	

CONTROL SYSTEM

6.1 Introduction

The SAS4A/SASSYS-1 control system model was developed for the design and analysis of control systems in LMR plants. The model is described in Ref. 6-1 while application of the model is described in Ref. 6-2. In this chapter, the model is described and guidelines for using the model are given. Essentially, the user should be able to set up an input deck and run the model using the material in this section and in Appendix 6.1.

The model is very flexible, allowing the user to select any number of plant variables for input to the control system as measured quantities. These signals can then be processed by a user defined network of mathematical blocks that implement the control equations. The output from these blocks can then be used to drive various actuators already existing in SAS4A/SASSYS-1 or they can be used to directly control plant variables in SAS4A/SASSYS-1. The model has a steady- state solution finder that can be used to determine initial values for demand signals and state variables that place the control system in a steady state that is consistent with the plant steady state as calculated by SAS4A/SASSYS-1. The control system model can also be used to calculate auxiliary variables and print their values.

The model is an integral component of SAS4A/SASSYS-1 and is accessed through the input deck in a manner similar to the other reactor component models. Before using the model however, one must write the mathematical equations that describe the desired plant control system and identify the plant variables that are to be measured and controlled. The user then transforms the equations and variables into a block diagram where the individual component blocks are basic mathematical elements such as integrators and summers. The input deck is prepared directly from this block diagram with each block definition occupying an input card and each plant variable that links with the control system also occupying an input card. Several other cards must also be entered to specify how the control system initial conditions are to be calculated and to assign values to parameters that control the accuracy and stability of the transient solution. A set of parameters also exists for controlling the printing of debug data. This output is useful for diagnosing input errors.

This section describes the basic model; it also gives some general guidelines for using the model. The section assumes the reader has a knowledge of power plant control systems and is able to write the equations that describe his system. The organization of the material is as follows. In Section 6.2 the general equation form that can be represented is given. It is very probable that the user's model fits this form but this should be verified. The solution techniques used to solve the block diagram equations are described in Section 6.3. Section 6.4 presents some general guidelines for selecting values of solution control parameters and describes some of the model features and how they are used. The input description is given in Appendix 6.1.

6.2 Generalized Model

The control system model was developed with the intent that a wide range of plant control systems can be simulated. For this purpose, two specific objectives were set. First, the model should be general enough to permit the user to assemble any set of control equations and specify how they interface to the plant solely through the input. And second, the model should employ a numerical method which is reliable in all foreseeable applications. Fulfilling these two goals led to the identification of a general equation form capable of representing all classes of plant control systems.

6.2.1 General Equations

The solution algorithms of the model are based on a general set of equations for the control system state variables and outputs. These equations are formulated under the assumption that the three components of a control system, the sensor, the controller, and the actuator, can all be modeled as ordinary differential equations. The general equation form is easy to deduce.

Since the sensor and actuator behavior are governed by physical laws and they are normally modeled in lumped parameter form, they are both described by

$$
\begin{aligned}
\frac{d}{dt}\underline{x}(t) &= \underline{f}\left(\underline{x}(t),\underline{u}(t)\right) \\
\underline{y}(t) &= \underline{g}\left(\underline{x}(t),\underline{u}(t)\right)
\end{aligned}
\tag{6.2-1}
$$

where

$\underline{x}(t)$ = n x 1 state vector;

$\underline{u}(t)$ = r x 1 input vector; and

$\underline{y}(t)$ = m x 1 output vector.

The controller also has the basic form of Eq. (6.2-1) as it consists of integrating and function elements. But in addition a derivative element is sometimes used in which case derivatives appear on the right hand side of Eq. (6.2-1). In practice the output signal from an integrator will be differentiated at most once so that the controller equation is

$$
\begin{aligned}
\frac{d}{dt}\underline{x}(t) &= \underline{f}\left(\underline{x}(t), \frac{d}{dt}\underline{x}(t),\underline{u}(t)\right) \\
\underline{y}(t) &= \underline{g}\left(\underline{x}(t),\underline{u}(t)\right).
\end{aligned}
\tag{6.2-2}
$$

The general equation form results when the equations for the three components are coupled and the signals that link to the plant are explicitly labeled

$$\frac{d}{dt}\underline{x}(t)=\underline{f}\left(\underline{x}(t),\ \frac{d}{dt}\underline{x}(t),\ \underline{u}_{mea}(t),\underline{u}_{dmd}(t)\right)$$
$$\underline{y}_{ctl}(t)=g\left(\underline{x}(t),\underline{u}_{mea}(t),\underline{u}_{dmd}(t)\right), \tag{6.2-3}$$

where

$\underline{u}_{mea}(t)$ = 1 x n_{mea} measured input vector;

$\underline{u}_{dmd}(t)$ = 1 x n_{dmd} demand input vector; and

$\underline{y}_{ctl}(t)$ = 1 x n_{ctl} control system output vector.

To guide the choice of initial conditions and their calculation for the above equations, we must consider the intended applications. Since the code is ultimately to be used for analysis of plant wide transients, the initial conditions must be compatible with the way in which these transients begin. Generally the user prescribes the plant steady state and therefore it should be reasonable to initialize the control system so that at time zero it preserves this steady state. In this case boundary conditions for the control system are taken from the plant, and control system time derivatives are set to zero. Writing the control equations explicitly in terms of the measured signals, control signals and the demand signals, Eq. (6.2-3) becomes

$$\underline{0}=\underline{f}\left(\underline{x}(0),\ 0,\underline{u}^{*}_{mea}(0),\ \underline{u}_{dmd}(0)\right)$$
$$\underline{0}=\underline{g}\left(\underline{x}(0),\ \underline{u}^{*}_{mea}(0),\underline{u}_{dmd}(0)\right)-\underline{y}^{*}_{ctl}(0), \tag{6.2-4}$$

where

$\underline{y}^{*}_{ctl}(0)$ = 1 x n_{ct1} vector of plant values associated with $\underline{y}_{ctl}(0)$; and

$\underline{u}^{*}_{mea}(0)$ = 1 x n_{mea} vector of plant values associated with $\underline{u}_{mea}(0)$.

The asterisk denotes steady state conditions in the plant. The initial conditions then that place the control system in steady state equilibrium with the plant are the values of $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$ that satisfy Eq. (6.2-4).

6.2.2 Block Diagram Approach

One might well ask what benefits can be obtained from a knowledge of this general equation. The principal benefit is a flexible modeling approach that permits the user to describe the plant control equations in a block diagram manner. The key to achieving this capability is the fact that the properties of the general equation form are well known and can be brought to bear on the development of a reliable numerical scheme.

The process by which the user describes his block diagram is analogous to the process of programming an analog computer. Basically, four types of information must be supplied. First, each mathematical block must be defined and the interconnections among them specified. Presently there are twenty-one blocks to choose from and these are shown in Fig. A6.1-1; additional blocks can be added if required. Each block can accept up to two signals at its input for processing and supply the result, termed a block signal, for further processing by other blocks. Second, each forcing function driving the collection of blocks must be defined. A demand signal is available for this purpose and references a user-supplied table of values that specifies the signal as a function of time. Third, plant measured quantities input to the collection of blocks must be defined. A measured signal is available for this purpose and permits access to a number of plant variables including temperature, flow, pressure and inventory in a number of reactor components. A complete list is given in Table A6.1-3. Finally, those block signals that are used to drive the plant must be defined. For that purpose, control signals can be defined by the user to represent, among other things, sources of external reactivity, feedwater mass flowrate and pump motor torque. A complete list is given in Table A6.1-3.

6.3 Solution Techniques

So far we have focused on the benefits derived from a generalized model, but have not touched on the methods used to implement the model. We will now describe the numerical techniques, first discussing the potential problems that can occur when solving a set of equations of the form Eq. (6.2-3) and then describing the numerical methods used to handle them.

Because the model is generalized, the solution techniques should be transparent for a wide range of situations that can arise. In the case of the steady-state solution finder, it is clear that the equations and variables to be solved for are given by Eq. (6.2-4). In certain instances these equations may not be square yet a solution exists while at other times the Jacobian of the right hand side may be singular. In the case of the transient solution it is important that the solution technique be able to maintain a user specified level of solution accuracy under a wide range of system response times. Solution techniques capable of handling these situations might be termed robust. We describe such techniques here.

The reader is cautioned that the difference equations we write may appear unconventional. The equations of reactor physics and thermal hydraulics when differenced in one-dimensional space appear with a single index denoting location in space. However in the generalized control system problem, geometry or space does not enter. Instead the continuum of space is replaced by a logical relationship among block outputs, block inputs and plant signals. These relationships must be recorded as part of the specific problem definition and, as will become clear in this work, vectors can be used to store these relationships. To summarize, the usual physical relationships that exist between points in space in conventional difference equations are replaced by logical relationships among blocks.

6.3.1 Definition of Block Diagram

The specification of the user's control equations and the solution of these are closely tied to the block diagram concept. Therefore, before the solution techniques can be discussed, we need to define the block diagram.

The block diagram is represented through several vectors whose entries define the types of blocks and the interconnections among blocks. The vectors are one-dimensional and the index to the elements can be thought of as analogous to a space index. To begin with, a unique signal number is assigned to the output of a block. If there are n_{blk} blocks occurring in the user's input and the *ith* one is assigned signal number m and the block is of type *k* then the following entries are created

$$L_{blk,i} = m \text{ and } K_{blk,m} = k \;;\; i = 1,\ldots,\, n_{blk}\,.$$

Further if the inputs to this block are signal *q* and *r* then the entries

$$J1_{blk,m} = q \text{ and } J2_{blk,m} = r$$

are also created. If the block is a non-dynamic one then the block operator is

$$F_k\left(S_{q,}^{j}\, S_{r}^{j}\right) = \text{value of signal } m \text{ at time } j$$

or, if the block is a dynamic one, then the block operator is

$$D_k\left(S_{q,}^{j}\, S_{r}^{j}\right) = \text{value of derivative of signal } m \text{ at time } j.$$

The variable S_q^j denotes that value of signal *q* at time *j*. The block operators are defined in Fig. A6.1-1. An auxiliary vector also stores information on dynamic blocks only. For the *ith* occurring dynamic block having signal number *m*, the entry

$$L_{dyn,i} = m; \qquad i = 1,\ldots,n_{dyn}$$

is created.

A unique signal number must also be assigned to each control signal. Recall a control signal is used to drive a plant variable and that the signal originates at the output of a block. If there are n_{ctl} control signals occurring in the user's input and the i^{th} one is assigned signal number *m*, then the following entries are created

$$L_{ctl,i} = m; i = 1, \ldots n_{ctl}.$$

Further, if this control signal is taken from the output of block *q*, then the entry

$$J_{ctl,m} = q.$$

is also created.

The vectors L_{blk}, K_{blk}, $J1_{blk}$ and $J2_{blk}$ thus define the block diagram. Both the steady-state and transient solution methods access the elements in these vectors to march through the block diagram in a manner analogous to the step-wise progression up an axial mesh that is used in one-dimensional thermal and hydraulic analysis codes. In the control system problem, however, a logical relationship among blocks is substituted for the spatial relationship among fluid cells that occurs in thermal hydraulic problems.

6.3.2 Steady-State Solution

The set of equations given by Eq. (6.2-4) are a set of non-linear equations. Having the boundary conditions $\underline{u}^*_{mea}(0)$ and $\underline{y}^*_{ctl}(0)$, we must solve this set of equations for the initial conditions $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$. However, before describing the solution method, we discuss two important considerations.

The first consideration is that the solution technique not require calculation of the inverse Jacobian of Eq. (6.2-4) with respect to the unknowns. Any non-linear equation solver that does will fail in certain cases when in fact a solution exists. If, for example, one of the unknowns feeds into a block that has a deadband region, then during iteration on the unknowns the input to this block may end up in the deadband zone. Then the derivative of the right-hand side of Eq. (6.2-4) will be zero with respect to the unknown, in which case the inverse of the Jacobian does not exist. The second consideration involves the relationship between the number of equations and the number of unknowns to be solved for. If the control system is properly designed then the equations are either square or they are over-determined. In the later case, the number of equations exceeds the number of unknowns. The solution method must be able to find a solution if it exists in either case. As for the non-linear equation solver used in this work, it will handle both the singular Jacobian and the over-determined equation situations. The solver is based on a least squares technique that is described in Ref. 6-3. When used in a case with deadband as described above, the equation solver will return a true solution if one exists. However, the returned solution may in fact be one of an infinite number.

The non-linear equation solver uses an iterative process to converge to the values of $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$ that satisfy Eq. (6.2-4). The equation solver provides successively more refined estimates for u_{dmd} and $\underline{x}(0)$ at the start of every iteration and expects the right-hand side of Eq. (6.2-4) to be calculated for each new set of values. Values for the right-hand side are generated by marching through the block diagram in the sequence defined by the vectors L_{blk}, K_{blk}, $J1_{blk}$ and $J2_{blk}$. The marching procedure is as follows. Suppose S_m^P denotes the value of signal m on the pth iterate. Assume that if the signal S_m^P is either a measured signal or a control signal that it has been set with the respective boundary condition contained in $u^*_{mea}(0)$ or $y^*_{ctl}(0)$. Assume also that if the signal S_m^P corresponds to one of the unknowns in $u_{dmd}(0)$ or x(0) that the equation solver has made its estimate for the signal prior to the p^{th} pass through the block diagram and the S_m^P has been set to this value. Then the value of the right hand-site of Eq. (6.2-4) for the p^{th} iteration is calculated using

$$i=1,\ldots n_{blk}$$

$$m=L_{blk,i};\quad k=K_{blk,m};\; q=J1_{blk,m};\; r=J2_{blk,m}$$

$$S_m^P=0,\;\; k=4$$

$$S_m^P=F_k\left(S_q^P,\, S_r^P\right),\;\; k=1,2,3,8\ldots21.$$

On completion the elements of f are

$$S^P_{J1_{blk,L_{dyn,i}}},i=1,\ldots n_{dyn}$$

and the elements of g are

$$S^P_{J_{ctl,L_{ctl,i}}},i=1,\ldots n_{ctl}.$$

6.3.3 Transient Solution

The numerical techniques used to solve Eq. (6.2-3) are based on explicit differencing and a numerical marching procedure. The numerical techniques have performed well for those problems examined to date. A time step control mechanism automatically adjusts time step size to maintain a specified level of accuracy. This usually results in a step size smaller than the time constant of the fastest component which is often a sensor. When the equations are stiff, other solution techniques may offer a computationally more efficient solution. However, experience has shown that the order of the control equations is usually small and that the computational demands of the current scheme are reasonable.

The block diagram is advanced across a time step in two phases. In the first step, the block signals are updated to the start of the time step. This involves setting the measured and demand signals and then marching through the block diagram while holding dynamic block signals constant

$$i=1,\ldots.n_{blk}$$

$$m=L_{blk,i};\quad k=K_{blk,m};\quad q=J1_{blk,m};\quad r=J2_{blk,m},$$

$$S_m^j=\frac{S_q^j-S_q^{j-1}}{t^j-t^{j-1}}\;,\quad k=4,$$

$$S_m^j=F_k\left(S_q^j,S_r^j\right),\;\; k=1,2,3,8,\ldots21$$

$$S_m^j=S_m^j,\qquad k=5,6,7.$$

Then in the second step, the block signals are advanced across the time step.

$$i=1,\ldots.n_{blk}$$

$$m=L_{blk,i};\quad k=K_{blk,m};\quad q=J1_{blk,m};\quad r=J2_{blk,m},$$

$$S_m^{j+1}=\frac{S_q^j-S_q^{j-1}}{t^j-t^{j-1}}\ ,\quad k=4,$$

$$S_m^{j+1}=F_k\left(S_q^{j+1},\ S_r^{j+1}\right),\ k=1,2,3,8,\ldots21$$

$$S_m^{j+1}=D_k\left(\frac{S_q^j+S_q^{j+1}}{2},\ \frac{S_r^j+S_r^{j+1}}{2}\right)\left(t^{j+1}-t^j\right)+S_m^j,\quad k=5,6,7.$$

On completion the elements of $\underline{x}^{j+1}$ are stored in

$$S_{L_{dyn,i}}^{j+1}\quad i=1,\ldots n_{dyn}.$$

An accurate and stable solution to both the control equations and the plant equations is obtained by controlling the basic time step size known as a subinterval. The initial size of a new subinterval is obtained by SAS4A/SASSYS-1 by extrapolating rates of change in the plant from the previous subinterval. The control equations are advanced first over this new subinterval according to the algorithm just described. Two time step control mechanisms can come into effect during integration of the control equations.

The first time step mechanism attempts to limit the error in the control equation solution that results from numerically integrating over the subinterval. An initial estimate for this error is made after the integration algorithm has obtained a solution at the end of the current subinterval. The estimate is made for each element of the vector x (i.e. dynamic blocks) by first estimating a value at the end of the current subinterval by linearly extrapolating the change across the previous subinterval:

$$S_{m,e}^{j+1}=S_m^j+\frac{S_m^j-S_m^{j-1}}{t^j-t^{j-1}}\left(t^{j+1}-t^j\right),\tag{6.3-1}$$

where

$$m=L_{dyn,i};\ i=1,\ldots n_{dyn}.$$

If S_m^{j+1} is the value calculated by the integration algorithm then the error estimate is

$$e_m^j=\frac{\left|S_m^{j+1}-S_{m,e}^{j+1}\right|}{\left|S_m^j\right|+F5SIG(m)}\tag{6.3-2}$$

where F5SIG(m) is the zero crossing parameter supplied by the user as discussed below. The solution has converged if the quantity e_m^j is less than the user-supplied value for the error criterion EPSCS. If the solution has not converged, then for purposes of control system integration only, the subinterval is bisected into two substeps and the control equations are again advanced over the subinterval. The error is again computed using Eq. (6.3-2) but using the value that resulted from the previous integration in place of $S_{m.e}^{j+1}$. If the subinterval is still not converged, it is again bisected so now there are four substeps in the subinterval. This process is repeated until the error between successive iterations as defined by Eq. (6.3-2) is less than the input value for EPSCS.

The second time step mechanism limits the relative change in the control solution over a single subinterval. Large and unrestricted changes can lead to instability between the control system solution and the plant solution. After the subinterval has converged as described above, the relative change in control signals is computed via

$$C_m^j = \frac{\left| S_m^{j+1} - S_m^j \right|}{\left| S_m^j \right| + F5SIG(m)} \tag{6.3-3}$$

where

$$m = L_{ctl,i};\ i = 1,\ldots n_{ctl}.$$

where F5SIG(m) is the zero crossing parameter whose value is supplied by the user. For the m that gives the largest value of C_m^j if this value C_m^j is greater than the user-supplied relative change criterion EPSCPL, then the subinterval time step is cutback so that the relative change EPSCPL is just met. The subinterval cutback size is obtained by linear interpolation so that the new size is the value of Δt that satisfies

$$\frac{\Delta t}{EPSCPL} = \frac{t^{j+1} - t^j}{C_m^j} \tag{6.3-4}$$

If the subinterval time step is cutback, then the control system integration starts over again using the new subinterval size.

A third time step mechanism is used to limit the relative change in the plant solution across a subinterval. This mechanism is analogous to the second time step mechanism and is described in Ref. 6-1.

6.4 Guide to User Application

This section provides some guidelines that should help tie together the model equations and solution techniques just described and the card input description given in Appendix 6.1. In this section signal definition rules that must be observed are stated, model capabilities are highlighted, and rules of thumb for choosing the values of solution control parameters are given.

6.4.1 Signal Definition Rules

Signals are defined through the input deck and the definitions must conform to certain rules. Any of the four signal kinds can appear anywhere in the signal card region of the input deck, subject to the following rules.

Rule 1 - A block signal that is output from other than an integrator, lag compensator or lead-lag compensator must have been previously defined in the input stream before it can be used as an input to another block. This rule is intended to avoid circular references and to maintain proper sequencing of signals during numerical integration.

Rule 2 - A demand signal or measured signal must pass through at least one block before it can be used as a control signal.

Rule 3 - Each signal must be assigned a unique signal number between 1 and 998.

The card format for defining a signal is given in Appendix 6.1.

6.4.2 Units

Generally all measured signals are in MKS units while all control signals should be calculated in these same units. The exceptions are those signals that are normalized to a steady state value; these are appropriately noted in Table A6.1-3.

The convention for demand signals is that demand tables are always entered by the user, normalized to a time zero value of unity. The actual value for a demand signal is calculated in the code by multiplying the current time entry in the demand table by the initial condition value. The next subsection describes how the initial condition value is obtained.

The units of a block output signal are determined solely by the units of the input signals and any conversion factors that are entered by the user as constants on the block definition card.

6.4.3 Initial Conditions

In order to begin a transient calculation, initial condition values are required for demand signals and for the integrator, lag compensator and lead-lag compensator blocks. There are basically three options available for setting these values. In the first option, all values are supplied by the user through input cards: in this case the steady state solution finder is bypassed. If the user is seeking the steady state solution, then a null transient may have to be run. In the second option, the steady state solution finder is used to solve for the steady state values. In the final option, a mixed set of initial conditions are used with some values read directly from the signal cards while the remaining values are solved for such that Eq. (6.2-4) is satisfied. The card input data required for each of these options is described below.

If the initial condition values are to be read from the input cards then the steady state solution finder should be bypassed by setting the J1SIG field on the 999 card to '0'.

Then the value for a demand signal and for the block signal of each integrator, lag compensator and lead-lag compensator is taken form the F4SIG field on the associated signal definition card.

If the initial condition values are to be calculated by the steady state solution finder then the J1SIG field on the 999 card is set to '1'. An initial guess for each demand signal and integrator initial condition variable must be supplied on the F4SIG field of the signal definition card. In addition the F3SIG field must be set to '0'. As a rule of thumb, the initial guess should be within 15% of the actual steady state value to ensure convergence. The lag compensator and lead-lag compensator are special cases and do not require initial condition information from the user.

Finally, if a mixed set of initial conditions is to be used, the card data is identical to the case directly above, except for those demand and integrator signals whose initial conditions are to be read from cards. For these signals the F3SIG field is set to '1.0' and the F4SIG field is set to the initial condition value desired.

6.4.4 Solution Accuracy

The control system modeling capability attempts to limit the solution error that is introduced during the numerical integration of the control equations over a subinterval. Recall the error is controlled by repeatedly bisecting the subinterval time step into substeps until integrating across the subinterval gives a relative error between successive iterations that is less than the user-supplied value for EPSCS. [The method was described in Section 6.3]. The value of EPSCS is input on a table card and occupies location 8001. A value of 0.01 is suggested for most applications.

6.4.5 Solution Stability

The modeling capability also attempts to maintain a stable solution to the coupled control system and plant equations. The basic idea is that stability is enhanced if the relative change in a control signal across a subinterval is maintained less than the user supplied value for EPSCPL. [The method was described in Section 6.3]. The value of EPSCPL is input on a table card and occupies location 8002. A value of 0.1 is suggested for most applications.

6.4.6 Zero Crossing Parameter

The zero crossing parameter in Eq. (6.3-2) is intended to prevent unnecessarily small time step size when a signal passes close to zero. The situation we seek to avoid occurs when the zero crossing parameter F5SIG is zero. Then the denominator in Eq. (6.3-2) is very small so that the relative error is very large. Time step size is severely reduced even though the absolute error in the signal may well be acceptably small. The solution is to control absolute error at the zero crossing and we do it through the relative error control mechanisms associated with Eq. (6.3-2) by proper choice of a value for F5SIG.

The appropriate value of F5SIG is problem dependent and is selected by the user for input to the code. The goal is to select a value that gives a desired level of absolute error near the zero crossing yet does not significantly impact the calculation of relative error away from the zero crossing. To do so we note that the code controls integration error using Eq. (6.3-2) so that on convergence the solution satisfies

$$\left| S_m^{j+1} - S_{m,e}^{j+1} \right| = EPSCS \left(\left| S_m^j \right| + F5SIG(m) \right) \tag{6.4-1}$$

where the value of *m* is restricted to those signals that are output by dynamic blocks. Near the zero crossing S_m^j will be insignificant so that Eq. (6.4-1) is equivalently

$$F5SIG(m) = \frac{\left| S_m^{j+1} - S_{m,e}^{j+1} \right|}{EPSCS}. \tag{6.4-2}$$

Note that the numerator is the absolute error in the solution at convergence. We can arrange for the numerator to take on a specific value by appropriately choosing the value of F5SIG(m) once the value of EPSCS has been selected. For example, suppose we want the absolute error on convergence near the zero crossing to be $S_m^o 10^{-4}$ where S_m^o is the maximum magnitude signal *m* is to take on over all time. If, the sake of illustration, a value of 10^{-2} was input for EPSCS, then we can achieve our absolute error objective by calculating the value of F5SIG(m) from Eq. (6.4-2),

$$F5SIG(m) = \frac{S_m^o 10^{-4}}{10^{-2}} = S_m^o 10^{-2}$$

Away from the zero crossing, the impact of F5SIG(m) is insignificant.

Similarly, the value of F5SIG(m) associated with a control signal should be selected as follows. The time step is adjusted down if necessary so that the largest relative change in a control signal is limited by Eq. (6.3-3) to

$$\left| S_m^{j+1} - S_{m,e}^j \right| = EPSCPL \left(\left| S_m^j \right| + F5SIG(m) \right) \tag{6.4-3}$$

where the value of m is restricted to those signals that are control signals. Near the zero crossing S_m^j will be insignificant so that Eq. (6.4-3) is equivalently

$$F5SIG(m) = \frac{\left| S_m^{j+1} - S_m^j \right|}{EPSCPL}. \tag{6.4-4}$$

Note that the numerator is the absolute change in the solution across the time step. We can arrange for the numerator to take on a specific value by appropriately choosing the value of F5SIG(m) once the value of EPSCPL has been selected. For example, suppose we want the absolute change in the control signal near the zero crossing to be as large as $S_m^o 10^{-3}$ before time step size is reduced. If, for the sake of illustration, a value of 10^{-1} was input for EPSCPL, then the absolute change objective will be met if F5SIG(m) is calculated from Eq. (6.4-4),

$$F5SIG(m) = \frac{S_m^o 10^{-3}}{10^{-1}} = S_m^o 10^{-2}$$

Away from the zero crossing, the impact of F5SIG(m) on the control of fractional change is insignificant.

REFERENCES

6-1. R. B. Vilim, Unpublished information, Argonne National Laboratory, 1986.

6-2. R. B. Vilim, T. Y. C. Wei, and F. E. Dunn, "Generalized Control System Modeling for Liquid-Metal Reactors," *Nucl. Sci. Eng.*, vol. 99, pp. 183-196, July, 1988.

6-3. J. J. More, "The Levenberg-Marquardt Algorithm: Implementation and Theory," *Proceedings of the Biennial Conference on Numerical Analysis*, Dundee, Scotland, June 28 - July 1, 1977, Springer-Verlag, New York, pp. 105-116, 1978.

APPENDIX 6.1
CONTROL SYSTEM INPUT DESCRIPTION

This appendix contains a description of the SAS4A/SASSYS-1 input block assigned to the control system model.

Input Block Structure

The input block structure is identical to the standard SAS4A/SASSYS-1 input block structure in all but one respect. A new card format known as a signal card has been introduced. These cards immediately follow the block identifier card and precede the standard data cards. The ordering of the different card types is depicted in the diagram below.

block identifier card

signal card # 1

signal card # 2

.

.

.

signal card # n

end of signal card

data card # 1

data card # 2

.

.

.

data card # m

block delimiter card

Block Identifier Card

Block identifier cards are described in Appendix 2.2. For the control system input block the number is 5 and the block name is INCONT.

Signal Cards

A signal card contains data fields for the Fortran variables

ISIG JTYPE J1SIG J2SIG F1SIG F2SIG F3SIG F4SIG F5SIG with the format descriptors 4I5, 5F10.3. These variables are defined in Table A6.1-1.

A signal card is used to define a signal in the user's block diagram. As described in the main body of this report there are four signal types: measured, demand, block and control. Each signal must be assigned a unique signal identification number using the ISIG field. The value of ISIG must lie between 1 and 998.

Measured Signal

A measured signal makes available to the block diagram the present value of a referenced SAS4A/SASSYS-1 variable. The correspondence between the variable that is referenced and the signal card data field values is given in Table A6.1-3. Note that all measured signals have a JTYPE value between -50 and -89.

Demand Signal

A demand signal makes available to the block diagram the product of the current value of a time dependent function defined by the user through a demand table and an initial condition value. A demand table is a set of ordered pair values supplied by the user in the format of Tables A6.1-2 and A6.1-4. The code obtains the demand signal value by linearly interpolating among the table entries using the current time. The initial value is obtained as described in Section 6.4. The correspondence between the demand table and the signal card data fields is given in Table A6.1-3. Note that a demand signal has a JTYPE value of -90.

Table A6.1-1. Signal Card Format

Column	Fortran Symbol	Definition	Variable Type
1	ISIG	Signal number	Integer
6	JTYPE	Signal type	Integer
11	J1SIG	Signal descriptor 1	Integer
16	J2SIG	Signal descriptor 2	Integer
21	F1SIG	Constant 1	Real
31	F2SIG	Constant 2	Real
41	F3SIG	Constant 3	Real
51	F4SIG	Constant 4	Real
61	F5SIG	Constant 5	Real

Table A6.1-2. Table Card Format

Column	Fortran Symbol	Definition	Variable Type
1	LOC	Storage location of VAR1	Integer
7	N	Number of consecutive locations	Integer
13	VAR1	Constant 1	Real
25	VAR2	Constant 2	Real
37	VAR3	Constant 3	Real
49	VAR4	Constant 4	Real
61	VAR5	Constant 5	Real

Table A6.1-3. Signal Cards

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Compressible volume pressure, PRESL3	-50	Volume number, ICV						
Measured	Liquid segment flowrate; FLOSL3	-51	Liquid segment number, ISGL						
Measured	Liquid cover gas interface elevation, ZINTR3	-52	Volume number, ICV						
Measured	Liquid mass, XLQMS3	-53	Volume number, ICV						
Measured	Cover gas volume, VOLGC3	-54	Volume number, ICV						
Measured	Time	-55							
Measured	Pump head, HEADP3	-56	Pump number, IPMP						
Measured	Liquid temperature, TLQCV3	-57	Volume number, ICV						
Measured	Liquid density, DNSCV3	-58	Volume number, ICV						
Measured	Wall temperature, TWLCV3	-59	Volume number, ICV						
Measured	Cover gas pressure, PRESG3	-60	Volume number, ICV						

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Cover gas mass, GASMS3	-61	Volume number, ICV						
Measured	Cover gas temperature, TGASC3	-62	Volume number, ICV						
Measured	Not used	-63							
Measured	Liquid segment temperature, TSLIN3	-64	Segment number, ISGL	Inlet=1 Outlet=2					
Measured	Pump speed, PSPED3	-65	Pump number, IPMP						
Measured	Core channel coolant flowrate, CHFL03	-66	Channel number, ICH	Inlet=1 Outlet=2					
Measured	Liquid node temperature, TLNOD3	-67	Node number, INOD						
Measured	Wall node temperature, TWNOD3	-68	Node number, INOD						
Measured	Liquid element temperature, TELEM	-69	Element number, IEL	Inlet=1 Outlet=2					
Measured	Not used	-70							
Measured	Core channel outlet temperature, CHFCOF	-71	Channel number, ICH	Inlet=1 Oulet=2					
Measured	Normalized reactor power, DEXP (POWVA (3,1))	-72							

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Normalized fission power, POWFSO * AMPO	-73							
Measured	Normalized decay heat $\sum_{i=1}^{POWDK} POWWT(i) \times POWDKH(i)$	-74							
Measured	Not used	-75,... -82							
Measured	Steam generator, feed-water mass flowrate in	-83		SG number					
Measured	Steam generator, feed-water enthalphy in	-84		SG number					
Measured	Steam generator, steam mass flowrate	-85		SG number					
Measured	Steam generator, steam temperature out	-86		SG number					
Measured	Steam generator, steam pressure	-87		SG number					
Measured	Steam generator, water level	-88		SG number					
Measured	Steam generator, steam enthalpy out	-89		SG number			Initial condi-tion flag	y_0	

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Demand	Demand table	-90	Demand table number	Number of entries in table				y_o	
Measured	Fuel Centerline Temperature	-101	Channel number	MZ mesh number	Scaling factor	Offset[c]			
Measured	Fuel Average Temperature	-102	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Fuel Surface Temperature	-103	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Clad Inner Wall Temperature	-104	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Clad Mid Wall Temperature	-105	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Clad Outer Wall Temperature	-106	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Coolant Temperature	-107	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Coolant Pressure	-108	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Coolant Saturation Temperature	-109	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Coolant Boiling Margin	-110	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Coolant Average Temperature	-111	Channel number	MZC mesh number	Scaling factor	Offset			

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Structure Inner Temperature	-112	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Structure Outer Temperature	-113	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Reflector Inner Temperature	-114	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Reflector Outer Temperature	-115	Channel number	MZC mesh number	Scaling factor	Offset			
Measured	Peak Fuel Temperature	-116	Channel number		Scaling factor	Offset			
Measured	Peak Clad Temperature	-117	Channel number		Scaling factor	Offset			
Measured	Peak Coolant Temperature	-118	Channel number		Scaling factor	Offset			
Measured	Minimum Boiling Margin	-119	Channel number		Scaling factor	Offset			
Measured	Coolant Inlet Temperature	-120	Channel number		Scaling factor	Offset			
Measured	Coolant Inlet Pressure	-121	Channel number		Scaling factor	Offset			
Measured	Coolant Inlet Flowrate	-122	Channel number		Scaling factor	Offset			
Measured	Coolant Outlet Temperature	-123	Channel number		Scaling factor	Offset			

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Coolant Outlet Pressure	-124	Channel number		Scaling factor	Offset			
Measured	Coolant Outlet Flowrate	-125	Channel number		Scaling factor	Offset			
Measured	Maximum Coolant Outlet Temperature	-126			Scaling factor	Offset			
Measured	Minimum Coolant Outlet Temperature	-127			Scaling factor	Offset			
Measured	Pin Bundle ΔT	-128	Channel number		Scaling factor	Offset			
Measured	Pin Bundle ΔP	-129	Channel number		Scaling factor	Offset			
Measured	Assembly Bundle ΔT	-130	Channel number		Scaling factor	Offset			
Measured	Assembly Bundle ΔP	-131	Channel number		Scaling factor	Offset			
Measured	Assembly Power	-132	Channel number		Scaling factor	Offset			
Measured	Linear Power	-133	Channel number	MZ mesh number	Scaling factor	Offset			
Measured	Peak Linear Power	-134	Channel number		Scaling factor	Offset			
Measured	Fission Gas Plenum Temperature	-135	Channel number		Scaling factor	Offset			

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Fission Gas Plenum Pressure	-136	Channel number		Scaling factor	Offset			
Block	Summer	1	Input signal 1, ISIG	Input signal 2, ISIG	g_1	g_2	g		
Block	Multiplier	2	Input signal 1, ISIG	Input signal 2, ISIG	g				
Block	Divider	3	Input signal 1, ISIG	Input signal 2, ISIG.	g				
Block	Differentiator	4	Input signal 1, ISIG		g				
Block	Integrator	5	Input signal 1, ISIG		g		Initial condi-tion flag	y_0	e_2^b
Block	Lag compensator	6	Input signal 1, ISIG		g	τ		y_0^a	e_2^b
Block	Lead-lag compensator	7	Input signal 1, ISIG		g	τ_1	τ_2	y_0^a	e_2^b
Block	Function generator	8	Input signal 1, ISIG	Function generator table number	g				
Block	Maximum	9	Input signal 1, ISIG	Input signal 2, ISIG					
Block	Minimum	10	Input signal 1, ISIG	Input signal 2, ISIG					

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Block	Time delay	11	Input signal 1, ISIG		τ			y_o[a]	
Block	Natural logarithm	12	Input signal 1, ISIG		g				
Block	Exponentiation	13	Input signal 1, ISIG	Input signal 2, ISIG	g				
Block	Velocity limiter	14	Input signal 1, ISIG	-	V_{down}	V_{up}	g		
Block	AND	15	Input signal 1, ISIG	Input signal 2, ISIG					
Block	OR	16	Input signal 1, ISIG	Input signal 2, ISIG					
Block	NOT	17	Input signal 1, ISIG						
Block	Comparator	18	Input signal 1, ISIG	Input signal 2, ISIG					
Block	Sample and hold	19	Input signal 1, ISIG	Input signal 2, ISIG					
Block	JK flip-flop	20	Input signal 1, ISIG	Input signal 2, ISIG				Q_o	
Block	Constant	21			g				
Block	Sine	22	Input signal 1, ISIG		g_1	g_2	g_3		

Signal Type	Variable	Card JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Control	Reactivity, $	-1	Signal number used						e_z^b
Control	Pump motor torque, normalized	-2	Signal number used	Pump number					e_z^b
Control	Steam generator, feedwater mass flowrate	-3	Signal number used	Steam generator number					e_z^b
Control	Steam generator, feedwater enthalpy	-4	Signal number used	Steam generator number					e_z^b
Control	Steam generator, steam mass flowrate	-5	Signal number used	Steam generator number					e_z^b
Control	Sodium valve loss coefficient	-6	Signal number used	Valve number					e_z^b
Control	Steam generator, steam pressure	-7	Signal number used	Steam generator number					e_z^b
Control	Air dump heat exchanger, air mass flowrate	-8	Signal number used	Air dump heat exchanger number					e_z^b

[a] Not required if steady state solution finder is used, J1SIG(999)=1.

[b] Zero crossing parameter.

[c] Measured channel signal result is Scaling Factor × Raw Value + Offset.

Table A6.1-4. Table Card Data

Location	Fortran Symbol	Definition/Comments
1	CTLTAB (J,J1SIG)	Table of normalized demand values. Dimension (20,100). Index J1SIG designates table number and J is element number in table.
2001	CTLTIM (J,J1SIG)	Times for CTLTAB table. Dimension (20,100).
4001	CTLFNC (J,J1SIG)	Table of function generator dependent variables. Dimension (20,100).
6001	CTLSIG (J,J1SIG)	Table of independent variables for CTLFNC table. Dimension (20,100).
8001	EPSCS	Convergence parameter for dynamic blocks over a subinterval.
8002	EPSCPL	Maximum relative change in a control signal over a subinterval.

Block Signal

A block signal makes available to the block diagram the value at the output of a block. The correspondence between the block characteristics and the signal card data fields is given in Table A6.1-3. Note that all block signals have a JTYPE value between 1 and 21. A measured, demand or block signal can be used as an input to a block by specifying on the block's signal definition card the signal identification number assigned to the input signal. The signals input to each block type are combined according to the mathematical expression given in Fig. A6.1-1.

Control Signals

A control signal is used to set the value of a SAS4A/SASSYS-1 variable equal to the value of a block signal. The correspondence between the block signal and the SAS4A/SASSYS-1 variable and the signal card data fields is given in Table A6.1-3. Note that all control signals have a JTYPE value between -1 and -8.

End of Signals

A sequence of signal definition cards is delimited by a signal card with the ISIG field entry equal to '999'.

This card also contains flags for the binary output file print interval and control of the stead state solution finder. First, the absolute value of the JTYPE field for the 999 card is sets the print interval for control system results output to the binary output file CONTROL.dat. Second, the J1SIG field is used to determine whether the steady state solution finder is to be used. An entry of '1' indicates that the steady state solution finder is to be used, while any other entry in this field causes the solution finder to be bypassed. (A discussion of the initial condition option is given in Section 6.4). Finally, the J2SIG field allows the user to control the amount of steady state output generated. An entry of '1' produces an extended output for trouble shooting purposes, while any other entry produces a standard output.

The JTYPE field is also used to generate an extended printout during the transient for debug purposes. The debug is generated by setting the JTYPE field of the 999 card to a negative value. The printout begins at the time specified on the F1SIG field.

Data Cards

A data card contains the data fields for the Fortran variables

LOC N VAR1 VAR2 VAR3 VAR4 VAR5

with the format descriptors 2I6, 5E12.5. The variables are defined in Table A.6.1-4.

A data card appearing in the control system block has a format identical to the standard SAS4A/SASSYS-1 data card used in all other input blocks and is processed in the same way. The format information given above is the same as in Chapter 2 and is given here for completeness.

Data cards are used to construct demand tables, function generator tables and to supply solution control parameters. These quantities and their storage locations are defined in Table A.6.1-4.

Sample Input

Figures A6.1-2 and A6.1-3 show examples of input for the control system.

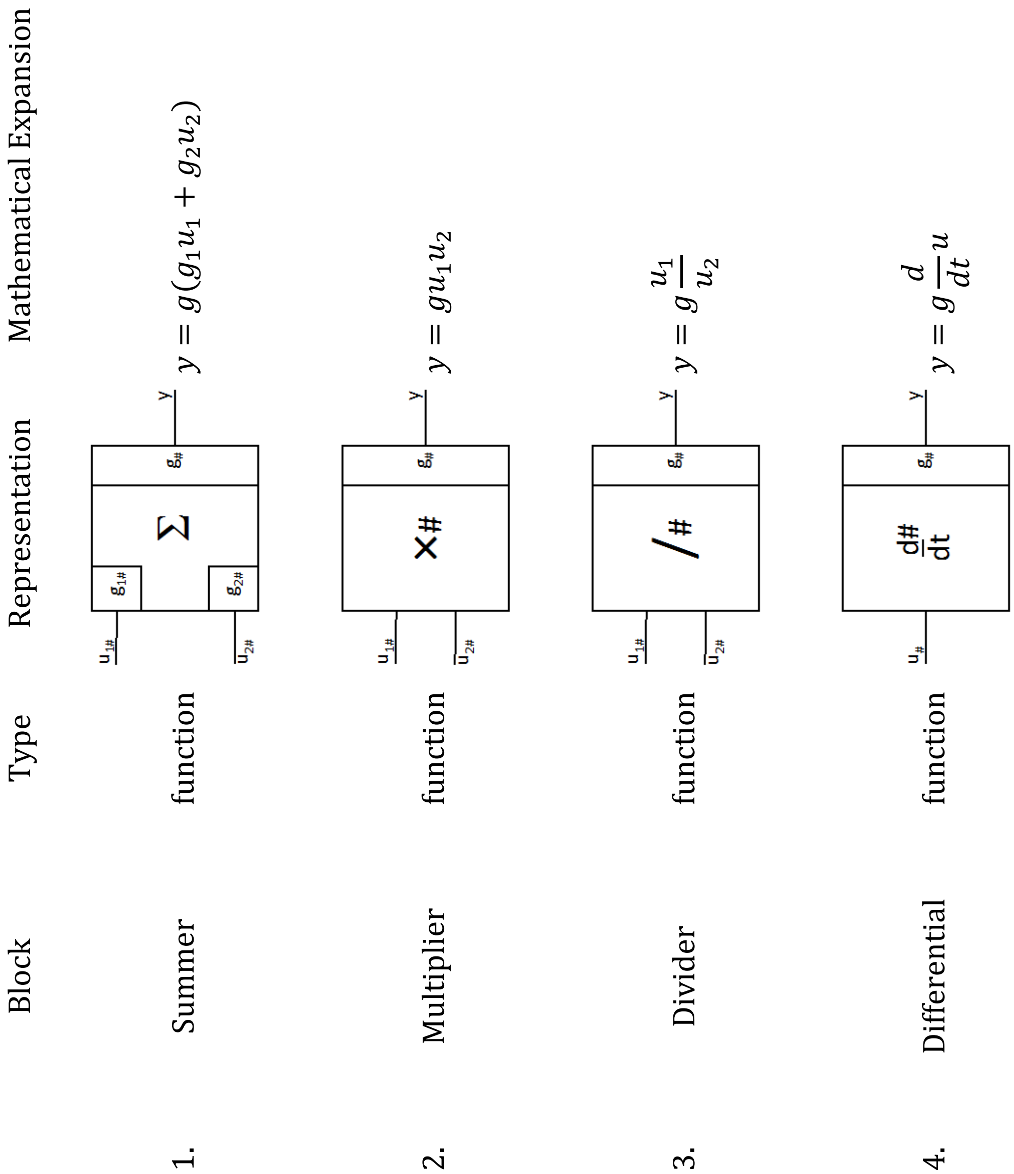


Figure A6.1-1. Basic Mathematics Blocks

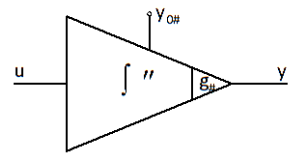
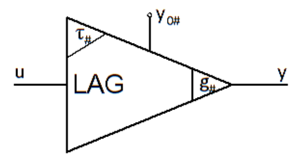
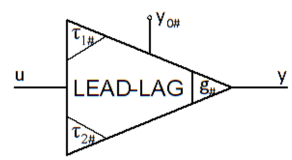
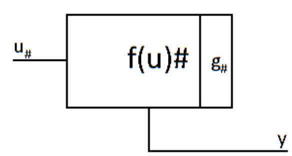
	Block	Type	Representation	Mathematical Expansion
5.	Integrator	dynamic		$y = y_0 + g \int_0^t u\, dt$
6.	Lag Compensator	dynamic		$y(0) = y_o$
7.	Lead-Lag Compensator	dynamic		$y + \tau_1 \frac{d}{dt} y = g(u + \tau_2 \frac{d}{dt} u)$ $y(0) = y_o$
8.	Function Generator	table		$y = g\, f(u)$

Figure A6.1-1. Basic Mathematical Blocks (Cont'd)

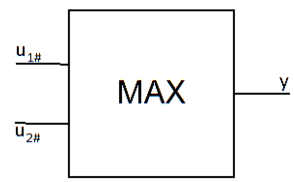
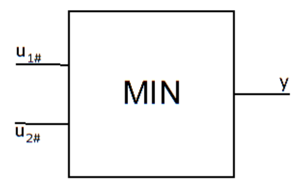
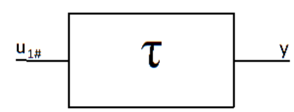
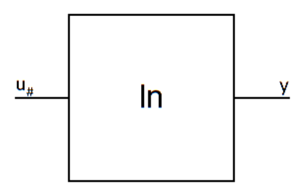
	Block	Type	Representation	Mathematical Expansion
9.	Maximum Value	function		$y = \max(u_1, u_2)$
10.	Minimum Value	function		$y = \min(u_1, u_2)$
11.	Time Delay	function		$y = u(0) \quad 0 \leq t \leq T$ $y = u(t - \tau) \quad t > T$
12.	Natural Logarithm	function		$y = \ln u$

Figure A6.1-1. Basic Mathematical Blocks (Cont'd)

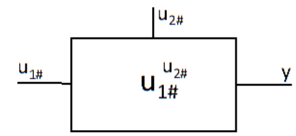
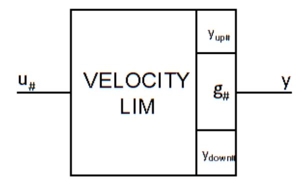
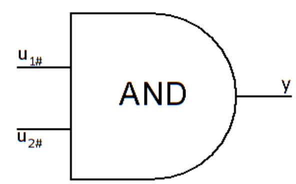
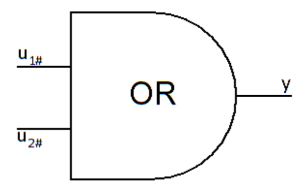
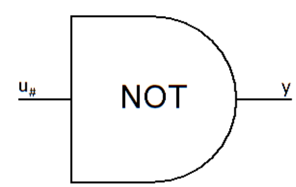
	Block	Type	Representation	Mathematical Expansion	
13.	Exponentiation	function		$y = u_1^{u_2}$	
14.	Velocity Limiter	function		$y = y_{down}$ $y = y_{up}$ $y = gu$ $y_{down} = y(t-h) - h\, v_{down}$ $y_{up} = y(t-h) + h\, v_{up}$	$gu < y_{down}$ $gu > y_{up}$ Otherwise
15.	AND	logic		$y = 1$ $y = 0$	$u_1 > 0, u_2 > 0$ otherwise
16.	OR	logic		$y = 0$ $y = 1$	$u_1 \leq 0, u_2 \leq 0$ otherwise
17.	NOT	logic		$y = 1$ $y = 0$	$u \leq 0$ $u > 0$

Figure A6.1-1. Basic Mathematical Blocks (Cont'd)

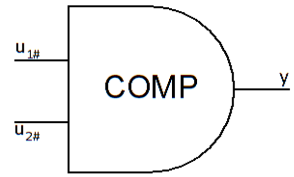
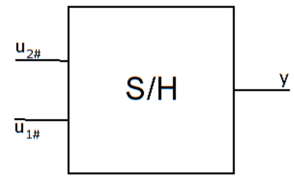
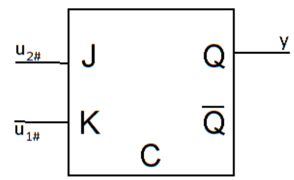
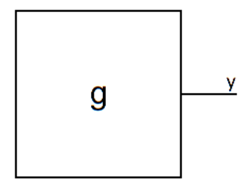
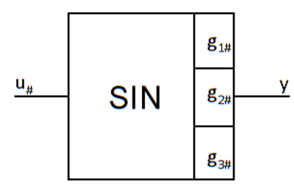
	Block	Type	Representation	Mathematical Expansion	
18.	Comparator	logic		$y = 0$ $y = 1$	$u_1 < u_2$ $u_1 \geq u_2$
19.	Sample and Hold	function		$y(t) = u_2(t)$ $y(t) = u_2(t_0)$	$u_1(t) \leq 0$ $u_1(t) \geq 0, t_o < t$ $u_1(t') \leq 0, t_{-1} \leq t' < t_0$
20.	J-K Flip Flop	logic		$y^{n+1} = Q^n$ $y^{n+1} = 0$ $y^{n+1} = 1$ $y^{n+1} = \bar{Q}^n$	$u_1 \leq 0, u_2 \leq 0$ $u_1 > 0, u_2 \leq 0$ $u_1 \leq 0, u_2 > 0$ $u_1 > 0, u_2 > 0$
21.	Constant	function		$y = g$	
22.	Sin	function		$y = g_1 \times \sin(g_2 u + g_3)$	

Figure A6.1-1. Basic Mathematical Blocks (Cont'd)

```
INCONT      5
#
#        JTYPE: Signal Type
#        |    J1SIG
#ISIG    |    |    J2SIG
#   |    |    |    |         F1SIG      F2SIG      F3SIG      F4SIG
#   |    |    |    |           |          |          |          |
    1  -57    1    0         0.0        0.0        0.0      100.0    # TL(CV1)
    2  -57    2    0         0.0        0.0        0.0      100.0    # TL(CV2)
    3  -57    3    0         0.0        0.0        0.0      100.0    # TL(CV3)
    4  -57    4    0         0.0        0.0        0.0      100.0    # TL(CV4)
    5    1    1    2      0.0245     0.7026        1.0      100.0    # 0.0245*TL(CV1)+0.7030*TL(CV2)
    6    1    3    4      0.2435     0.0294        1.0      100.0    # 0.2430*TL(CV3)+0.0294*TL(CV4)
    7    1    5    6         1.0        1.0        1.0      100.0    # TLBAR
    8    8    7    1         1.0        0.0        0.0  1000000.0    # Table of K_orf
    9   -6    8    1         0.0        0.0        0.0  10000000.    # K_orf for Valve 1
   10    8    7    2         1.0        0.0        0.0       0.01    # Table of Reactivity
   11   -1   10    0         0.0        0.0        0.0       0.01    # Control Rod Reactivity
#
  999
#
#                        Function Generator 1: Independent Values
#                          |            |            |            |            |
  4001      5      1.0E10       1.0E10       1.0E9        1.0E6        1.0E3
  4006      2         0.0          0.0
#
#                        Function Generator 1: Dependent Values
#                          |            |            |            |            |
  6001      5         0.0        738.5       739.0        739.5        740.0
  6006      2       740.5      10000.0
#
#                        Function Generator 2: Independent Values
#                          |            |            |            |            |
  4021      5         0.0          0.0        0.01         0.05         0.10
  4026      2      0.1954       0.1954
#
#                        Function Generator 2: Dependent Values
#                          |            |            |            |            |
  6021      5         0.0        738.5       739.0        739.5        740.0
  6026      2       740.5      10000.0
#
#                        EPSCS: Convergence parameter for dynamic blocks
#                          |            EPSCPL: Maximum relative change in a control
#                          |            |        signal over a subinterval
  8001      2         0.1          0.5
    -1
```

Figure A6.1-2. Sample Input for the Control System

```
INCONT     5
#
#        JTYPE: Signal Type
#        |    J1SIG
#ISIG    |    |    J2SIG
#   |    |    |    |         F1SIG     F2SIG     F3SIG
#   |    |    |    |         |         |         |
    1 -120    1    0       1.8    -459.67            # Coolant Inlet Temperature, °F
    2 -123    1    0       1.8    -459.67            # Coolant Outlet Temperature, °F
#
#   |    |    |    |         |         |         |
    3    1    1    2       1.0       1.0       0.5   # Summer/Average, °F
#
  999
#
#                      EPSCS: Convergence parameter for dynamic blocks
#                      |           EPSCPL: Maximum relative change in a control
#                      |           |       signal over a subinterval
  8001     2         0.1         0.1
    -1
```

Figure A6.1-3. Sample Input for the Control System