ANL-FRA--160

DE87 014429

ANL-FRA-160

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

A USER'S GUIDE TO THE SASSYS-1 CONTROL SYSTEM MODELING CAPABILITY

by

R. B. Vilim

Applied Physics Division

June 1987

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

A USER'S GUIDE TO THE SASSYS-1 CONTROL SYSTEM MODELING CAPABILITY

by

R. B. Vilim
Applied Physics Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

ABSTRACT

This report describes a control system modeling capability that has been developed for the analysis of control schemes for advanced liquid metal reactors. The general class of control equations that can be represented using the modeling capability is identified, and the numerical algorithms used to solve these equations are described. The modeling capability has been implemented in the SASSYS-1 systems analysis code. A description of the card input, a sample input deck and some guidelines for running the code are given.

I. INTRODUCTION

This report serves as a user's guide to a control system modeling capability that has been developed for the analysis of control schemes for advanced liquid metal reactors. The capability can be interfaced to any standard reactor systems code; here we describe it as interfaced to the SASSYS-1 reactor systems code. The class of control equations that can be solved is presented and the solution algorithms are described. A description of the card input, a sample input deck, and some general guidelines for running the code are given. For a brief description of power plant control systems, the reader is referred to Reference 1. For an illustration of how the modeling capability can be used for design and analysis of plant control systems, the reader is referred to Reference 2.

The modeling capability 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 SASSYS-1 or they can be used to directly control plant variables in SASSYS-1. The modeling capability 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 SASSYS-1. The control system modeling capability can also be used to calculate auxiliary variables and print their values.

The control system model is an integral component of SASSYS-1 and is effected through the input deck in a manner similar to the other reactor component models. To use the model, one must first 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 report is organized as follows. Section II introduces the general class of control equation that can be represented. 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 III. Section IV presents some general guidelines for selecting values of solution control parameters and describes some of the modeling capability features and how they are used. The input description is given in Appendix A.

II. GENERALIZED MODELING CAPABILITY

The control system modeling capability was developed with the intent that a wide range of plant control systems could be simulated. For this purpose, two specific objectives were set. First, the modeling capability 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 modeling capability 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.

A. General Equation Form

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 since they are normally modeled in lumped parameter form, they are both described by

$$\begin{aligned} \frac{d}{dt}\underline{x}(t) &= \underline{f}(\underline{x}(t), \underline{u}(t)) \\ \underline{y}(t) &= \underline{g}(\underline{x}(t), \underline{u}(t)) \end{aligned} \qquad (1)$$

where

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

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

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

The controller also has the basic form of Eq. (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. (1). In practice the output signal from an integrator will be differentiated at most once so that the controller equation is

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

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

$$\begin{aligned} \frac{d}{dt}\underline{x}(t) &= \underline{f}(\underline{x}(t), \frac{d}{dt}\underline{x}(t), \underline{u}_{mea}(t), \underline{u}_{dmd}(t)) \\ \underline{y}_{ctl}(t) &= \underline{g}(\underline{x}(t), \underline{u}_{mea}(t), \underline{u}_{dmd}(t)), \end{aligned} \qquad (3)$$

where

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

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

$\underline{y}_{ctl}(t) \;=\; 1 \times n_{ctl}$ 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 demand signals, Eq. (3) becomes

$$\begin{aligned} \underline{0} &= \underline{f}(\underline{x}(0),\, \underline{0},\, \underline{u}^{*}_{mea}(0),\, \underline{u}_{dmd}(0)) \\ \underline{0} &= \underline{g}(\underline{x}(0),\, \underline{u}^{*}_{mea}(0),\, \underline{u}_{dmd}(0)) - \underline{y}^{*}_{ctl}(0), \end{aligned} \qquad (4)$$

where

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

$\underline{u}^{*}_{mea}(0) \;=\; 1 \times n_{mea}$ 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. (4).

B. Block Diagram

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; a particular signal type is available for each kind. First, the user's forcing functions that drive the collection of blocks must be defined. A demand signal as a function of time is provided as a table. Second, the plant measured quantities that also drive the collection of blocks must be defined. A measured signal is available for this purpose. Access is permitted to a number of plant variables including temperature, flow, pressure and inventory in a number of reactor components. Third, the mathematical blocks must be defined and the interconnections among them specified. Each block can accept up to two signals at its input for processing to yield the result, termed a block signal, for further processing. Finally, those block signals that are used to drive the plant must be defined. For that purpose, a control signal whose value is taken from the output of a block can be defined by the user to drive, among other things, control rods, feedwater mass flowrate and pump motor torque.

The block diagram, and hence functions $\underline{f}$ and $\underline{g}$, are 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 each block signal. If there are n_{blk} block signals occurring in the user input, and if the ith block signal is assigned signal number m and corresponds to a block of type k, then the following entries are created:

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

Further, if the inputs to this block are signals q and r then the entries

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

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

$$F_k(S_q^j, S_r^j) = \text{value of signal m at time j},$$

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

$$D_k(S_q^j, S_r^j) = \text{value of derivative of signal m at time j}.$$

The variable S_q^j denotes the value of signal q at time j. A complete list of blocks is given in Fig. A.1. An auxiliary vector also stores information cn 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 that 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 input, and if the ith control signal is assigned signal number m, then the following entries are created:

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

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 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 typical assembly thermal hydraulics analysis codes. In the control system problem, however, a logical relationship among blocks is substituted for the spatial relationship that exists among fluid cells in thermal hydraulics problems.

III. SOLUTION TECHNIQUES

So far we have focused on the benefits derived from a generalized modeling capability, 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. (3) and then describing the numerical methods used to handle them.

Because the modeling capability 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. (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.

A. Steady-State Solution

We must solve the steady state equations given by Eq. (4) for the initial conditions $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$ given the boundary conditions $\underline{u}^*_{mea}(0)$ and $\underline{y}^*_{ctl}(0)$. The numerical solution of these equations is relatively straightforward; essentially, any non-linear equation solver can be used to solve them. The basic approach is to provide initial estimates for the unknowns $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$ and then refine these estimates through successive iterations so that the right-hand side of Eq. (4) tends to zero.

The solution method thus requires us to calculate the right-hand side of Eq. (4) given estimates for $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$. To do so we march through the block diagram to obtain values for $\underline{f}$ and $\underline{g}$. Suppose S_m^p denotes the value of signal m at the start of the pth iteration. 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 in the associated element of $\underline{u}_{mea}^*(0)$ or $\underline{y}_{ctl}^*(0)$. Assume that if another signal S_m^p is associated with one of the elements of $\underline{u}_{dmd}(0)$ or $\underline{x}(0)$, that an estimate for the start of the pth iteration has been assigned. Then the value of the right-hand side of Eq. (4) for the pth iteration is calculated as follows: for i = 1,..., n_{blk} set

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

and then based on the value of k calculate one of the following:

$$S_m^p = 0, \qquad k = 4,$$

$$S_m^p = F_k(S_q^p, S_r^p), \qquad k = 1,2,3,8,\ldots, 21$$

For the pth iteration, the elements of $\underline{f}$ are

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

and the elements of $\underline{g}$ are

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

The right-hand side of Eq. (4) is calculated using these values of $\underline{f}$ and $\underline{g}$.

The solution strategy is to minimize the length of the right-hand side of Eq. (4) by iterating on the values of $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$. The search procedure is described in Reference 3. Basically, we supply the value of the right-hand side at the start of each iteration and from this the search strategy decides on the appropriate adjustments to $\underline{u}_{dmd}(0)$ and $\underline{x}(0)$ for the next iteration. A solution has been found when the right-hand side of Eq. (4) becomes zero.

The solution search strategy of Reference 3 was selected for its ability to handle both singular Jacobians and over-determined systems. First, non-linear equation solvers that rely on calculation of the inverse of the Jacobian of the right-hand side of Eq. (4) with respect to the unknowns will fail in certain cases. If, for example, one of the unknowns feeds into a block that has a deadband region, iterating on the unknown may place the input to this block in the deadband zone. Then the derivative of the right-hand side of Eq. (4) will be zero with respect to the unknown, in which case the inverse Jacobian does not exist. Second, the solution method must be able to handle both square and over-determined systems. Typically, the equations are square, that is, the number of equations equals the number of unknowns. However in some instances the equations may be over-determined, that is, the number of equations exceeds the unknowns.

One would normally expect a solution to Eq. (4) to exist, but not necessarily a unique solution. With the plant at steady-state, as it is in our case, there should be a set of demand signal values that yield control signal values that are consistent with the plant steady-state. However, if the control system contains deadband, then the solution may not be unique. The search strategy of Reference 3 will alert the user when the solution has the property that it can be perturbed in a particular direction to yield another solution, as would likely be the case when deadband is present.

B. Transient Solution

The numerical techniques used to solve Eq. (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 reader is cautioned that the notation used to represent the differenced control equations may appear unconventional, compared to the equations of reactor physics and thermal hydraulics. These latter equations when differenced in one-dimensional space appear with a single index denoting location in space. However, geometry or space is not pertinent here. Instead, the continuum of space is replaced by a logical relationship among control equation blocks. The relationship among blocks in a specific problem is given by the vectors L_{blk}, K_{blk}, $J1_{blk}$ and $J2_{blk}$ as described in subsection II.B.

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

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

and then based on the value of k calculate one of the following:

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

$$S_m^j = F_k(S_q^j, S_r^j), \qquad k = 1,2,3,8,\ldots,21,$$

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

Then in the second step, the block signals are advanced across the time step. For $i = 1,\ldots,n_{blk}$ set

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

and then based on the value of k calculate one of the following:

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

$$S_m^{j+1} = F_k(S_q^{j+1}, S_r^{j+1}), \qquad k = 1,2,3,8,\ldots,21;$$

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

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

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

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 SASSYS 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 $\underline{x}$ (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:

$$S_{m,e}^{j+1} = S_m^j + \frac{S_m^j - S_m^{j-1}}{t^j - t^{j-1}} (t^{j+1} - t^j), \qquad (5)$$

where

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

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

$$e_m^j = \frac{|S_m^{j+1} - S_{m,e}^{j+1}|}{|S_m^j| + F5SIG(m)} \tag{6}$$

where F5SIG(m) is the zero crossing parameter supplied by the user as discussed in subsection IV.F. 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) 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) 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

$$C_m^j = \frac{|S_m^{j+1} - S_m^j|}{|S_m^j| + F5SIG(m)} \tag{7}$$

where

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

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 of C_m^j is greater than the user-supplied relative change criterion EPSCPL, then the subinterval time step is cut back 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

$$\frac{\Delta t}{EPSCPL} = \frac{t^{j+1} - t^j}{C_m^j} . \tag{8}$$

If the subinterval time step is cut back, 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 Reference 1.

IV. A GUIDE TO USER APPLICATION

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

A. 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 A.

B. 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 A.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.

C. 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. (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 from 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 '0.0'. 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.

D. 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 III.B.] The value of EPSCS is input on a table card and occupies location 8001. A value of 0.01 is suggested for most applications.

E. 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 III.B.] The value of EPSCPL is input on a table card and occupies location 8002. A value of 0.1 is suggested for most applications.

F. Zero Crossing Parameter

The zero crossing parameter in Eq. (6) 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) 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) 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) so that on convergence the solution satisfies

$$|S_m^{j+1} - S_{m,e}^{j+1}| = \text{EPSCS}\ (|S_m^j| + \text{F5SIG}(m)) \tag{9}$$

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. (9) is equivalently

$$\text{F5SIG}(m) = \frac{|S_m^{j+1} - S_{m,e}^{j+1}|}{\text{EPSCS}} \ . \tag{10}$$

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, for 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. (10),

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

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 size is adjusted down if necessary so that the largest relative change in a control signal is limited by Eq. (7) to

$$|S_m^{j+1} - S_m^j| = \text{EPSCPL}\ (|S_m^j| + \text{F5SIG}(m)) \tag{11}$$

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. (11) is equivalently

$$F5SIG(m) = \frac{|S_m^{j+1} - S_m^j|}{EPSCPL}. \tag{12}$$

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. (12),

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

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

V. SUMMARY

This report serves as a user's guide to a control system modeling capability that has been developed for the analysis of control schemes for advanced liquid metal reactors. The plant control equations are shown for all practical applications to belong to a particular class of mathematical equations. The numerical techniques used to solve these equations are described. The user models the control system in a block diagram manner, assembling particular equations through the input by connecting together the basic mathematical blocks that are available. The modeling capability has been implemented in the SASSYS-1 systems analysis code. A description of the card input is given along with a sample input deck.

VI. REFERENCES

1. R. B. Vilim, "A Generalized Control System Model for the SASSYS-1 Systems Analysis Code," FRA-TM-154, Argonne National Laboratory, April 1987.

2. R. B. Vilim, T. Y. C. Wei, F. E. Dunn, "Generalized Control System Modeling for Liquid Metal Reactors," to be submitted to Nuclear Science and Engineering, June 1987.

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 A
CONTROL SYSTEM INPUT DESCRIPTION

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

A.1 Input Block Structure

The input block structure is identical to the standard 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

A.2 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 A.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.

A.2.1 Measured Signal

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

A.2.2 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. The independent variable is time and the dependent variable is to be normalized to a time zero value of unity. The values are entered through a table card defined in Table A.2. The SASSYS input storage locations for demand table data are given in Table A.4. The code generates the demand signal value by linearly interpolating among the table entries using the current time. The initial value is obtained as described in Section IV.C. The correspondence between the demand table and the signal card data fields is given in Table A.3. Note that a demand signal has a JTYPE value of -90.

A.2.3 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 A.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. A.1.

A.2.4 Control Signals

A control signal is used to set the value of a SASSYS-1 variable equal to the value of a block signal. The correspondence between the block signal and the SASSYS variable and the signal card data fields is given in Table A.3. Note that all control signals have a JTYPE value between -1 and -7.

A.2.5 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 control of the steady state solution finder. First, 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 subsection IV.C.) Secondly, 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.

A flag also exists for generating an extended printout during the transient for debug purposes. The debug is generated by setting the JTYPE field to '1'. The printout begins at the time specified on the F1SIG field.

A.3 Data Cards

A data card contains the data fields for the Fortran variables

LOC N VAR1 VAR2 VAR2 VAR3 VAR4 VAR5

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

A data card appearing in the control system input block has a format identical to the standard SASSYS-1 data card used in all other SASSYS-1 input blocks and is processed in the same way. The format information given above is the same as in the SASSYS-1 manual 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.4.

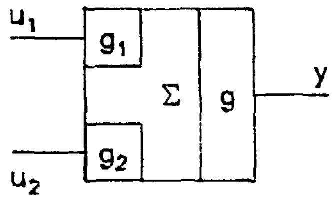
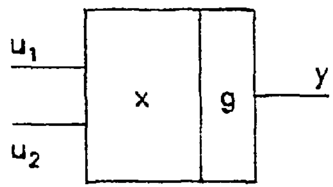
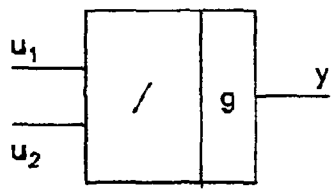
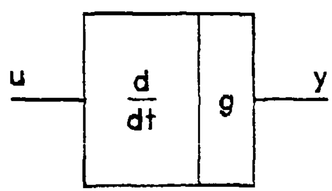
	Block	Type	Representation	Mathematical Expression
1.	summer	function		$y = g(g_1u_1 + g_2u_2)$
2.	multiplier	function		$y = gu_1u_2$
3.	divider	function		$y = g\frac{u_1}{u_2}$
4.	differentiator	function		$y = g\frac{d}{dt}u$

Fig. A.1. Mathematical Blocks

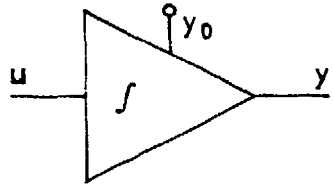
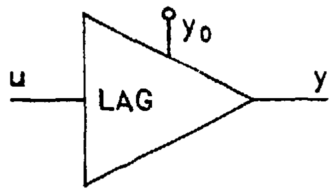
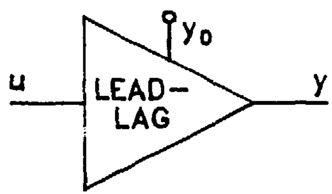
Block	Type	Representation	Mathematical Expression
5. integrator	dynamic		$y = y_0 + g \int_0^t u\, dt$
6. lag compensator	dynamic		$y + \tau \frac{d}{dt} y = g u$ $y(0) = y_0$
7. lead–lag compensator	dynamic		$y + \tau_1 \frac{d}{dt} y = g(u + \tau_2 \frac{d}{dt} u)$ $y(0) = y_0$

Fig. A.1. Mathematical Blocks (Contd.)

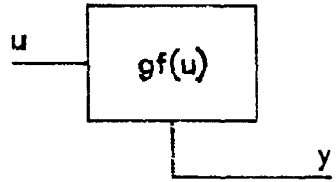
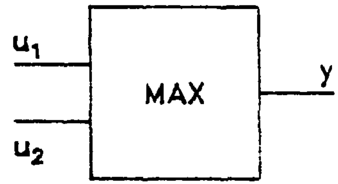
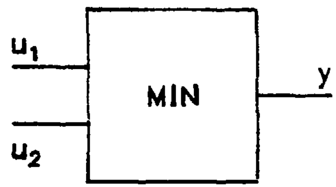
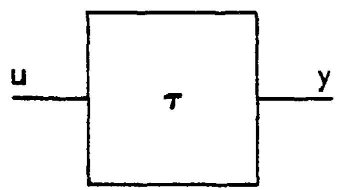
	Block	Type	Representation	Mathematical Expression	
8.	function generator	table		$y = gf(u)$	
9.	maximum value	function		$y = max(u_1, u_2)$	
10.	minimum value	function		$y = min(u_1, u_2)$	
11.	time delay	function		$y = y_0$ $y = u(t - \tau)$	$0 \leq t \leq T$ $t > T$

Fig. A.1. Mathematical Blocks (Contd.)

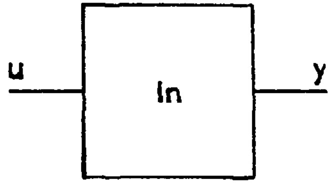
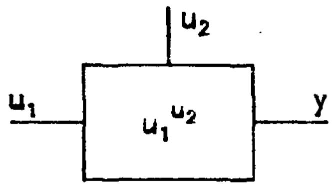
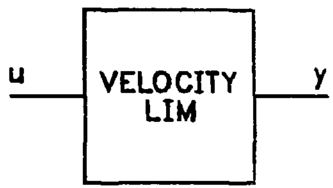
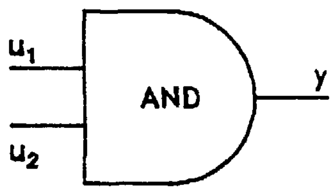
Block	Type	Representation	Mathematical Expression
12. natural logarithm	function		$y = \ln u$
13. exponentiation	function		$y = u_1^{u_2}$
14. velocity limiter	function		$y = y_{down}$ $gu < y_{down}$ $y = y_{up}$ $gu > y_{up}$ $y = gu$ otherwise $y_{down} = y(t - h) - hv_{down}$ $y_{up} = y(t - h) + hv_{up}$
15. AND	logic		$y = 1$ $u_1 > 0, u_2 > 0$ $y = 0$ otherwise

Fig. A.1. Mathematical Blocks (Contd.)

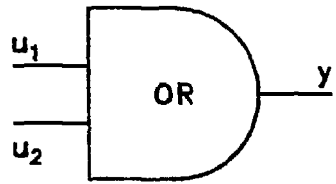
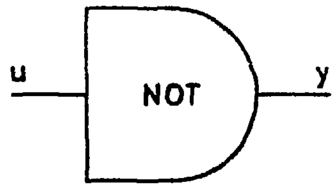
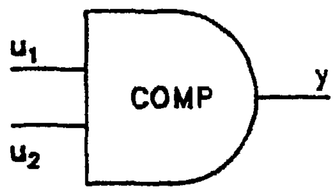
	Block	Type	Representation	Mathematical Expression	
16.	OR	logic		$y = 0$ $y = 1$	$u_1 \leq 0, u_2 \leq 0$ otherwise
17.	NOT	logic		$y = 1$ $y = 0$	$u \leq 0$ $u > 0$
18.	comparator	logic		$y = 0$ $y = 1$	$u_1 < u_2$ $u_1 \geq u_2$

Fig. A.1. Mathematical Blocks (Contd.)

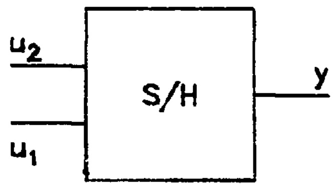
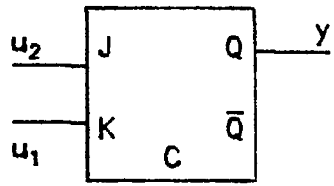
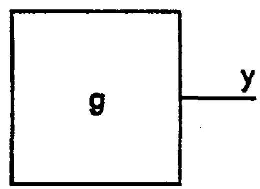
Block	Type	Representation	Mathematical Expression
19. sample and hold	function		$y(t) = u_2(t) \quad u_1(t) \leq 0$ $y(t) = u_2(t_0) \quad u_1(t) \geq 0,\ t_0 < t$ $u_1(t') \leq 0,$ $t_{-1} \leq t' < t_0$
20. J–K flip flop	logic		$y^{n+1} = Q^n \quad u_1 \leq 0,\ u_2 \leq 0$ $y^{n+1} = 0 \quad u_1 > 0,\ u_2 \leq 0$ $y^{n+1} = 1 \quad u_1 \leq 0,\ u_2 > 0$ $y^{n+1} = \bar{Q}^n \quad u_1 > 0,\ u_2 > 0$
21. constant	function		$y = g$

Fig. A.1. Mathematical Blocks (Contd.)

TABLE A.1. Signal Card Format

Column	Format Code	Definition
1	I5	Signal number
6	I5	Signal type
11	I5	Signal descriptor 1
16	I5	Signal descriptor 2
21	F10.3	Constant 1
31	F10.3	Constant 2
41	F10.3	Constant 3
51	F10.3	Constant 4
61	F10.3	Constant 5

TABLE A.2. Table Card Format

Column	Format Code	Definition
1	I6	Storage location of VAR1
7	I6	Number of consecutive locations
13	E12.5	Constant 1
25	E12.5	Constant 2
37	E12.5	Constant 3
49	E12.5	Constant 4
61	E12.5	Constant 5

TABLE A.3. Signal Data

Signal		Card Entries[a]							
Type	Variable	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						
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, CHFLO3	-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						

TABLE A.3. Signal Data (Contd.)

Signal		Card Entries[a]							
Type	Variable	JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
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 Outlet=2					
Measured	Normalized reactor power, DEXP (POWVA (3,1))	-72							
Measured	Normalized fission power, POWFSO * AMPO	-73							
Measured	Normalized decay heat $\sum_{i=1}^{NPOWDK} POWWT(i) \times POWDKH(i)$	-74							
Measured	Not used	-75,...,-82							
Measured	Steam generator, feedwater mass flowrate in	-83		SG number					
Measured	Steam generator, feedwater enthalpy 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					

TABLE A.3. Signal Data (Contd.)

Signal		Card Entries[a]							
Type	Variable	JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Measured	Steam generator, steam enthalpy out	-89		SG number			Initial condition flag	y_0	
Demand	Demand table	-90	Demand table number	Number of entries in table				y_0	
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 condition flag	y_0	ε_z^c
Block	Lag compensator	6	Input signal 1, ISIG		g	τ		y_0^b	ε_z^c
Block	Lead-lag compensator	7	Input signal 1, ISIG		g	τ_1	τ_2	y_0^b	ε_z^c
Block	Function generator	8	Input signal 1, ISIG	Function generator table number	g				
Block	Maximum	9	Input signal 1, ISIG	Input signal 2, ISIG					

TABLE A.3. Signal Data (Contd.)

Signal		Card Entries[a]							
Type	Variable	JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Block	Minimum	10	Input signal 1, ISIG	Input signal 2, ISIG					
Block	Time delay	11	Input signal 1, ISIG		τ			y_0^b	
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_0	
Block	Constant	21			g				
Control	Reactivity, $	-1	Signal number used						ε_z^c

TABLE A.3. Signal Data (Contd.)

Signal		Card Entries[a]							
Type	Variable	JTYPE	J1SIG	J2SIG	F1SIG	F2SIG	F3SIG	F4SIG	F5SIG
Control	Pump motor torque, normalized	-2	Signal number used		Pump number				ϵ_Z^c
Control	Steam generator, feedwater mass flowrate	-3	Signal number used		Steam generator number				ϵ_Z^c
Control	Steam generator, feedwater enthalpy	-4	Signal number used		Steam generator number				ϵ_Z^c
Control	Steam generator, steam mass flowrate	-5	Signal number used		Steam generator number				ϵ_Z^c
Control	Sodium valve loss coefficient	-6	Signal number used		Valve number				ϵ_Z^c
Control	Steam generator, steam pressure	-7	Signal number used		Steam generator number				ϵ_Z^c

[a]Format codes: 4I5, 5F10.3.
[b]Not required if steady state solution finder is used, J1SIG(999)=1.
[c]Zero crossing parameter.

TABLE A.4. Table 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.

APPENDIX B
SAMPLE INPUT DECK

The card input listed below was used to control mass flowrate in a primary and intermediate loop. Each controller measures the flowrate in its respective loop and compares this with a demand signal. The error signal is integrated and the output of the integrator is used to adjust the motor torque driving a pump.

```
INCONT     5     1     0
    1  -51     8     0        0.          0.          0.          0.          0.
    2  -90     1     4        0.          0.          0.          0.          0.
    3  -90     2     2        0.          0.          0.          0.          0.
    4    2     2     3       1.0          0.          0.          0.          0.
    5    1     1     4       1.0       -1.0-4.365E-04             0.          0.
    6    5     5     0       1.0          0.          0.          0.          0.
    7   -2     6     1        0.          0.          0.          0.          0.
    8  -51     9     0        0.          0.          0.          0.          0.
    9  -90     3     4        0.          0.          0.          0.          0.
   10  -90     4     2        0.          0.          0.          0.          0.
   11    2     9    10        1.          0.          0.          0.          0.
   12    1     8    11        1.       -1.0-4.390E-04             0.          0.
   13    5    12     0        1.          0.          0.          0.          0.
   14   -2    13     2        0.          0.          0.          0.          0.
  999    0     0     0        0.          0.          0.          0.          0.
    1      4        1.0         1.0         1.3         1.3
  201      4         0.        10.0        20.0      1000.0
   21      2     2291.7      2291.7
  221      2         0.       1000.
   41      4        1.0         1.0         1.3         1.3
  241      4         0.        10.0        20.0       1000.
   61      2      2278.       2278.
  261      2         0.       1000.
  801      2      1000.       1000.
   -1
```