INTERACTIVE SIMULATION LANGUAGE FOR HYBRID COMPUTERS*

R. D. Benham and G. R. Taylor
Interactive Mini Systems, Inc., 5312 W Tucannon
Kennewick, Washington 99336

ABSTRACT

Interactive Simulation Language (ISL) now offers a powerful and easy to use software system for hybrid computers by providing:

- Uniform programming for analog and digital
- Exportable software to different digitals
- Variable time scale selection for digital
- Automatic documentation of analog voltages
- Automatic set-up of analog pots and DCA's
- Automatic verification of analog patchboards.

This paper illustrates the programming procedures and operational conveniences for solving a second order differential equation a variety of ways using all digital, non-synchronous hybrid, synchronous hybrid, and all analog techniques.

INTRODUCTION

The ISL system provides a large number of operational elements such as found on analog computers. These elements include integrators, adders, multipliers, and many others as described in Appendix 1. An ISL program is formed by "interconnecting" these elements to satisfy the equations being solved. Thus, for hybrid programming the digital computer is an extension of the analog computer. The exceptions to uniformity are:

- ISL eliminates scaling
- ISL provides bipolar outputs
- ISL has more math elements

There are many advantages for using uniform (or nearly uniform) methods and symbols for hybrid work:

- It is easier to visualize the math model
- Digital programming is simplified
- Digital operations are performed efficiently
- Large problems can be solved on small computers.

Programming techniques for all digital, hybrid, and all analog methods are illustrated with the mass-spring-damper equation:

$$M \frac{d^2y}{dt^2} + B \frac{dy}{dt} + K y = 0$$

where,

$M = 2$ (mass)
$B = 1.5$ (damping coefficient)
$K = 2$ (spring constant)
$y(0) = 5$ (initial displacement)
$\frac{dy}{dt}(0) = 0$ (initial velocity)

ALL DIGITAL SOLUTION

The ISL (all digital) simulation diagram for the example is shown in Figure 1. Notice the absence of sign change in the integrator and that scaling is required for DAC and ADC units. Since ISL executes each math block in sequence, it is good programming procedure to form the derivatives before integrating. Figure 1 shows that the POT's forming the second derivative are evaluated before their outputs are used by the integrator.

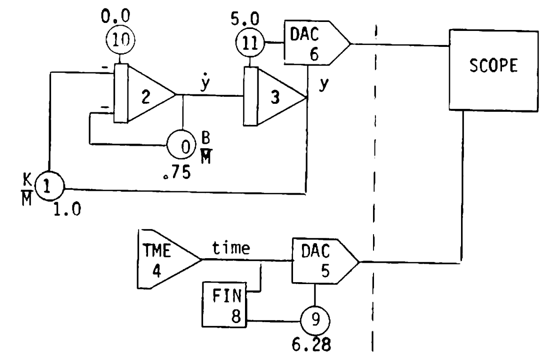

Figure 1. All Digital Solution

The ISL program is "patched up" interactively at the console keyboard (teletype) by using the math definitions given in Appendix 1 and the commands of Appendix 2. The ISL listing showing the interconnections of Figure 1 is:

```
MASS SPRING DAMPER

0   POT  2, 7.5E-1
1   POT  3, 1.0E 0
2   INT  10, -0, -1
3   INT  11, 2
4   TME  4
5   DAC  4, 9, 0
6   DAC  3, 11, 1
7   CCC  AN EXAMPLE COMMENT
8   FIN  4, 9
9   CON = 6.28E 0
10  CON = 0.0E 0
11  CON = 5.0E 0
12  END  12
```

A normal sequence for entering the program is:

- Erase the user program area with the "K" command
- Enter the program title (ie. MASS SPRING DAMPER) with the "J" command
- Append the user program as shown in lines 0 to 12 with the "A" command.

ISL has an on-line compiler with error checking and diagnostics. Therefore most keyboard errors are easily corrected. Corrections at a later time can also be made with the MODIFY "M" command.

The ISL listing merely shows the interconnection of blocks. For example, line 2 shows that the integrator receives an IC input from line 10 and negative inputs from lines 0 and 1. Notice that DAC outputs (5 and 6) require scaling information. For example line 5 shows DAC channel 0 outputs block 4 normalized to block 9.

The program is run by using the commands in Appendix 2. By typing "I" and a "carriage return", the console printer will respond by asking for the integration step size (S) and the printing interval (T). Since no printing is required the value for the "T" entry is ignored.

Each ISL math element is executed in sequence until the FIN block is reached (block 8). If the FIN statement is not satisfied (ie. block 4 is less than block 9) control is transferred to block 0 for another iteration through the program. When the FIN statement is satisfied (block 4 is equal or greater than block 9) the next block is evaluated until the END statement is reached. END transfers control back to the ISL monitor.

The computing process can be interrupted and ISL placed in HOLD mode for reading out component values, changing parameter values, modifying the program, changing integration step size, etc. The computation process can then be continued if desired. Notice that ISL has operational modes of IC, HOLD, and COMPUTE that are similar to analog computer modes.

When one wishes to save an ISL user program on tape or disk he merely types the appropriate argument with the "W" command from the ISL monitor. Programs can be re-entered with the "E" command. ISL user programs written to paper tape are interchangeable with all other 16 bit computers for which ISL is available. Thus, ISL provides "exportable" user software.

ALL DIGITAL SOLUTION WITH ANALOG POTENTIOMETER

The damping coefficient can be varied with a hand set potentiometer by replacing line 0 with a MUL and line 7 with an ADC command. Figure 2 shows the modified simulation diagram.

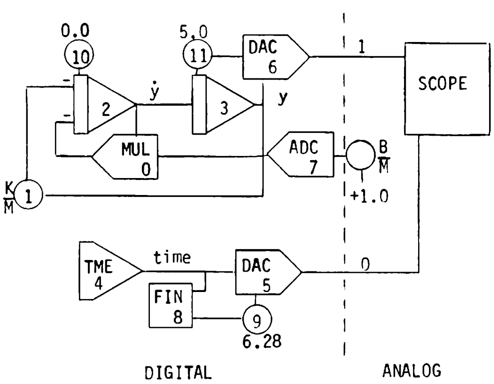

Figure 2. Parameter Change With Analog Pot

To view repetitive solutions an automatic change mode (CHM) instruction is added to the end of the program. Thus, when the CHM instruction is executed, the mode is changed from IC to COMPUTE or from COMPUTE to IC and a new iteration sequence is started. The modified ISL listing for this example is:

```
MASS SPRING DAMPER

0   MUL  2, 7
1   POT  3, 1.0E 0
2   INT  10, -0, -1
3   INT  11, 2
4   TME  4
5   DAC  4, 9, 0
6   DAC  3, 11, 1
7   ADC  0
8   FIN  4, 9
9   CON = 6.28E 0
10  CON = 0.0E 0
11  CON = 5.0E 0
12  CHM  12
13  END  13
```

This program is run as explained in the section on all digital solutions. Note in the all digital mode ISL allows communication with DAC and ADC units without otherwise controlling the modes of the analog computer.

NON-SYNCHRONOUS HYBRID SOLUTION

On large problems it is not uncommon to run out of analog computing elements. When this happens the user must:

- Reduce the problem complexity
- Acquire more analog computing elements
- Use the analog and digital computers together.

Frequently it is desired to use the digital computer for algebraic calculations such as function generation, multiplication, division, log, sin, etc. For illustration purposes we will use the analog computer for integration and the digital computer for the second derivate calculation. Figure 3 is the simulation diagram.

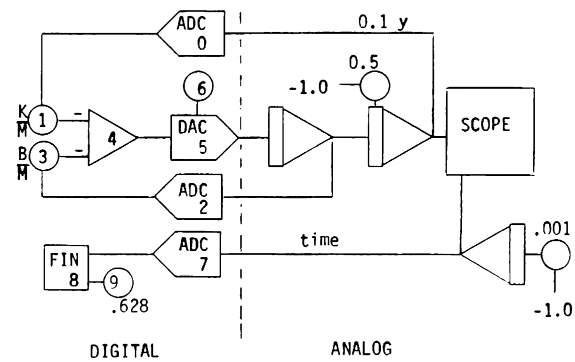

Figure 3. Second Derivative Calculation on Digital

Notice that it is necessary to start both computers at the same time. However, it is not necessary to control the iteration rate of the digital computer. The digital calculations occur at maximum rate. The analog computer is time scaled to allow a stable solution. The ISL listing for this hybrid program is:

```
HYBRID MASS SPRING DAMPER

0   ADC  0
1   POT  0, 1.0E 0
2   ADC  1
3   POT  2, 7.5E-1
4   ADD  -1, -3
5   DAC  4, 6, 0
6   CON = 9.999E-1
7   ADC  2
8   FIN  7, 9
9   CON = 6.28E-1
10  CHM  10
11  END  11
```

Non-synchronous operation is specified from the hybrid monitor (see Appendix 2) by selecting a clock frequency of zero (T = 0). The simulation must be controlled from the console keyboard. This represents no restrictions on the use of the analog computer because ISL has full address selection for voltage read-out and coefficient modification (see "R" command in Appendix 2). In addition the ISL hybrid monitor provides automatic methods (with single commands) to:

- Save all pot and DCU settings to tape (W)
- Reset all pot and DCU settings from tape (E)
- List any or all analog component groups on the console printer (or line printer) (L)
- Automatically save static test conditions for setting up and checking out the analog computer (see D and V commands)
- Analog mode selection with the M command

Analog console selection (see H command from the ISL monitor).

SYNCHRONOUS HYBRID OPERATION

When it is desired to integrate on both computers, it is necessary to clock the frame time of the digital computer. The "I" command (Appendix 2) shows that it is necessary to enter the time step (S), the clock frequency (T), and specify the time scale factor of the digital computer. If it is physically possible for the computer to accept this information, the computation process proceeds normally. If impossible, an error message is given.

The simulation diagram for synchronous operation is shown in Figure 4 and the ISL listing follows.

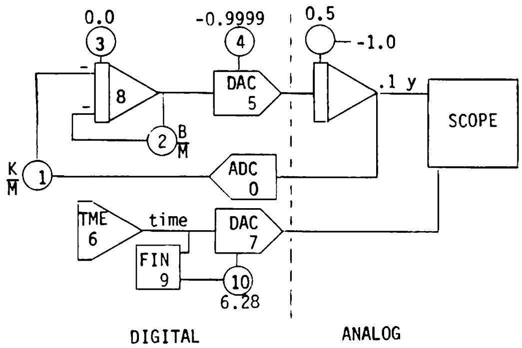

Figure 4. Synchronous Hybrid Operation

SYNCHRONOUS MASS SPRING DAMPER

```
0   ADC  0
1   POT  0, 1.0E 0
2   POT  8, 7.5E-1
3   CON = 0.0E 0
4   CON =-9.999E-1
5   DAC  8, 4, 0
6   TME  6
7   DAC  6, 10, 1
8   INT  3, -1, -2
9   FIN  6, 10
10  CON = 6.28E 0
11  CHM  11
12  END  12
```

As with previous examples, the console keyboard offers complete control over both computers.

ALL ANALOG WITH AUTOMATIC POT SETTING

Figure 5 shows the scaled analog simulation diagram. The pots for this simulation can be hand set using the "R" command from the ISL Hybrid Monitor or they can be automatically set with an ISL program using the COF block as indicated in Figure 5.

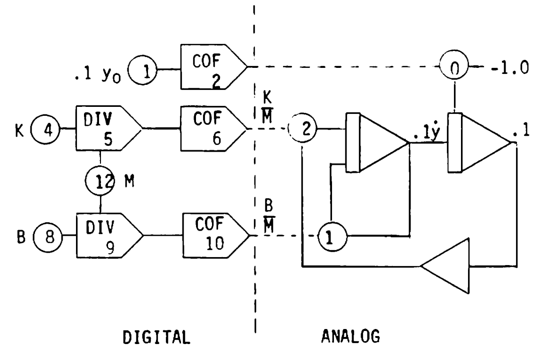

Figure 5. Automatic Pot Setting

The pot setting program is convenient for changing a parameter that appears in several pot settings (such as M). It is a simple matter to change the value of M (line 12) with the "P" command and then run the pot setting program. The ISL hybrid monitor can be used for controlling modes if desired, or the problem can be run using the normal controls on the analog computer. The ISL listing for the pot setting program is:

POT SET PROGRAM

```
0   CCC  SET IC FOR Y
1   CON = 5.0E-1
2   COF  1, 0
3   CCC  SET K/M
4   CON = 2.0E 0
5   DIV  4, 12
6   COF  5, 2
7   CCC  SET B/M
8   CON  1.5E 0
9   DIV  8, 12
10  COF  9, 1
11  CCC  VALUE OF M
12  CON = 2.0E 0
13  END  13
```

SPEED AND COMPUTING CAPACITY

The largest (known) problem solved with ISL utilized 462 statements. This required 3820 memory locations. Since the ISL program requires about 5K of memory, this large problem was solved with an 8K EAI 640 digital computer. With 32K of memory a problem over 10 times larger can be solved on the digital computer. The breakdown of the 462 math elements used is given in the table below:

CON	POT	MUL	SQT	DIV	ADD	FNG	EXP	INT	TDL	Misc.
70	73	94	9	44	107	9	6	23	5	22

The computing time required to execute all 462 blocks was 0.15 sec. More modern computers have up to 1/2 the cycle times of the EAI 640 computer and thus can process simulations up to twice as fast.

APPENDIX 1

MATHEMATICAL INSTRUCTIONS

ARITHMETIC

function	instruction		remarks
absolute value	ABS	N1	R = |N1|
add subtract	ADD	±N1, ±N2,... ±N9	R = ±N1 ± N2 ± ... N9
constant	CON	Nk	R = Nk
divide	DIV	N1, N2	R = N1/N2
equate	EQT	N1, N2	R = O : N2 = N1
multiply	MUL	N1, N2	R = N1 x N2
pot	POT	N1, y.yyyE y	R = N1 x y.yyyE y

TRANSCENDENTALS

function	instruction		remarks	
sine	SIN	N1	R = sin N1	radians
	SND	N1	R = sin 2π N1	degrees
cosine	COS	N1	R = cos N1	radians
	CSD	N1	R = cos 2π N1	degrees
logarithm	LOG	N1	R = log(10) N1	
	LNX	N1	R = log(e) N1	
exponential	TXP	N1	R = (10) exp N1	
	EXP	N1	R = (e) exp N1	
square root	SQT	N1	R = square root N1	

SUBROUTINE

function	instruction		remarks
subroutine	SBR	N1, N2 . . . N8	N1 = block transfer no. N2 . . . N8 are inputs SBR can reference all blocks except 0 to 10 in the main program
return	RTN	K	return to block N + 1 of main program from SBR K is meaningless

INPUT/OUTPUT

function	instruction		remarks
heading	HDR	T1, . . . T5	label TTY/L.P. columns with up to 6 chrs. each
print/plot	PRI	N1, . . . N5	print/plot to TTY/L.P.
display* (scope)	DIS	N1, N2, N3, N4	X - axis = N1/N2 Y - axis = N3/N4
digital* to analog conv	DAC	N1, N2, K	R = N1/N2 K = channel no.
analog* to digital conv	ADC	K	R = ADC output K = channel no.

CONTROL

function	instruction		remarks
change mode	CHM	K	reverses mode IC to C or C to IC
count	CNT	K	when iteration count = K skip next instr and reset count to K
end	END	K	reverse mode when in I and go to block O otherwise terminate comp
finish	FIN	N1, N2	If N1 ≥ N2 take next instr otherwise up-date time and go to block O
go to	GTO	Nk	go to block Nk
transfer	TFR	N1, N2, N3	if N1 ≥ N2 go to N3

LOGICAL

function	instruction		remarks	
input relay	IRL	N1, N2, N3, N4	R = N3	N1 ≥ N2
			R = N4	otherwise
limiter	LIM	N1, N2, N3	R = N1	N3 > N1
			= N3	N1 ≤ N3 ≤ N2
			= N2	N3 < N2
step size change	SCH	N1, N2, N3, K	if N1 ≥ N2	S = N3 and T = K

SPECIAL FUNCTIONS

function	instruction		remarks
comment*	CCC	EXAMPLE COMMENT	one line of comment
function generation	FNG	N1	R = f(N1)
increment	INC	N1, N2	R = ∑N2 + N1 = N1 in I C mode
integrate	INT	N1, N2, . . . N9	R = ∫(N2 + . . . N9)dt + N1
variable time delay	TDL	N1, N2, N3	R = f(N1)(T - 20N3/N2) N3 ≥ S · N2 = f(N1)(T - 20S) N3 < S · N2
time	TME		each iteration advances time by S

APPENDIX 2. MONITOR COMMANDS

ISL MONITOR (*)		ISL HYBRID MONITOR ($)	
A	Append program		
B	Back to numerical listing from plot		
C	enter Compute mode		
		D	Dump analog voltages to mass storage device
E	Enter program from external device	E	Enter pot settings from mass storage device
Fn,m	edit line m of FNG n		
Hn	select analog console n and go to Hybrid monitor →		
I	set initial conditions by setting the time step (S) and the printing interval (T)	I	Set Initial Conditions. The step size (S) is entered as usual. However the printing entry (T) is used to select the exponent of the desired clock frequency as follows: T = 0 Computations are not clocked but analog modes are controlled by ISL T = 1 Exponents of the clock pulse rate. = 2 After terminating this entry, the printer types = 3 'TIME SCALE = 1.000E 0// = 4 The user can accept real time by typing a carriage return, or a new factor is entered.
J	enter Job (program) title		
K	Kill (erase) current program		
L	List current program	Ln	List address and output voltage n = 0 All addresses and voltages n = 1 Amplifiers and Multipliers n = 2 Pot and DCU outputs n = 3 Trunks n = 4 Coefficient settings n = 5 Function generators
Ln	List line n		
Ln,m	List lines n through m		
Mn	Modify line n	M	Select analog Mode. The computer responds by typing MODE = The user enters H, I, O, or P for HOLD, IC, OPERATE, or POTSET.
Mn,m	Modify lines n through m		
N	print highest line Number		
P	Parameter read and/or set		
Q	interator selection (Quadrature) n = 0 EULER n = 1 RUNGE-KUTTA 2 n = 2 RUNGE-KUTTA 4		
R	print Remaining memory locations	R	Read analog components. The user enters the analog address and the computer prints the output voltage. Examples are: A003 = <u>0.5612</u> (computer types underlined value) C001 = <u>0.7585</u>// (type CR if coefficient is ok, if not enter new coefficient)
Un	select output Unit n = 0 teletype n = 1 line printer	Un	Select output Unit n = 0 teletype n = 1 line printer
V	read numerical Value	V	Verify previously saved data (prepared with the D command) against current analog values. If the difference is greater than ± 0.0025 the selected printer will print the address and the old and new values.
W	Write program to external device	W	Write pot coefficients to mass storage device
space	Define a CON with current line no.		
.	exit APPEND mode	.	← Return to ISL monitor

NOTE: NUMERICAL ENTRIES ARE TERMINATED WITH A SPACE, COMMA, OR CARRIAGE RETURN (CR).