Monograph:

Linkens, D. A., Gray, L.S., Menad, M et al. (1 more author) (1981) Psicon-A Simulation Package for the Design of Digitally Controlled Systems. Research Report. ACSE Report 162 . Department of Control Engineering, University of Sheffield, Mappin Street, Sheffield

[redacted]

PSICON - A Simulation package for the design of digitally controlled systems

by

D.A. Linkens, L.S. Gray, M. Menad, N. Mort*

Department of Control Engineering,
University of Sheffield,
Mappin Street, Sheffield S1 3JD.

*Control Systems Section,
Royal Naval Engineering College,
Manadon.

Research Report No. 162.

October 1981

PSICON - A Simulation package for the design of digitally controlled systems

D.A. Linkens, L.S. Gray, M. Menad, N. Mort*

Department of Control Engineering, University of Sheffield.
*Control Systems Section, R.N. Eng. College, Manadon.

ABSTRACT

A simulation language is described which is based on a block-oriented continuous systems simulator called PSI. Using the message transfer capability of the multi-tasking operating system of a mini-computer, external digital control algorithms can be developed rapidly in Fortran, while retaining the advantages of a flow-chart based and precompiled structure of the continuous simulation language. The use of the package is illustrated with two examples. The first is an anaesthetics problem where time-delay dynamics are controlled using an external Fortran programme comprising a Smith predictor plus a PID regulator. The second example is a self-tuning adaptive autopilot for ship steering. The package gives fast simulation and good interactive properties, with the external controller segments giving the required flexibility for a wide range of uses.

1. Introduction

Simulation languages for the study of continuous systems dynamical behaviour fall largely into two categories - equation-oriented and block-oriented. In the Department of Control Engineering, Sheffield, examples of each of these approaches are available on the large multiuser interactive Perkins-Elmer 3220 minicomputer. Limitations in each of these packages have led to the development of the language described in this paper.

Equation-oriented simulators describe relationships between the variables in the process by means of a set of equations. The syntax of the language usually allows the inclusion of most mathematical operators and hence great flexibility is possible in the range of systems which can be simulated. This is particularly true when Fortran segments are allowed within the system description. Equation-oriented languages require pre-processing before the simulation can be run. This involves phases such as translation, sorting and compiling. The version available at Sheffield is a derivative of CSMP and the timing problems associated with the multistage pre-processing led to interest in alternative approaches to simulation. The main problem is that the smallest change in the problem equations necessitates the whole preprocessor to be re-run. This is very time consuming and mitigates against good interactive simulation. This is particularly frustrating to those familiar with analogue computing and its associated hands-on facility for programme changes and variable read-out.

In contrast, block-oriented simulators have a structure similar to an analogue computer approach. The problem is prepared in a block diagram format familiar to control engineers, and is based on an analogue-type flow diagram. The problem is entered into the language directly from the flow-chart. The whole language is held in the computer in compiled form so that simulations can be run as soon as all the necessary parameters have been set up. Thus changes in the simulation structure or parameters can be very easily incorporated and the problem re-run. In this way good interactive properties are obtained. To allow for as wide a range of blocks as possible and relatively large problem size the precompiled package is large in terms of memory allocation. In the case of the Perkins-Elmer 3220 this has been no major limitation. The main disadvantage, however, is the lack of flexibility in incorporating special functions or segments. Everything must be describable in terms of standard blocks. Apart from being inefficient in terms of the number of blocks being used for quite simple functions, in many cases of modern control methods the algorithms cannot be described in such a way.

The above considerations have prompted the development of PSICON. The major section is the block-oriented language PSI with its good interactive facilities. The extension to it is the use of multi-tasking to allow for the inclusion of Fortran sub-routines to give the increased flexibility for incorporation of modern control algorithms. Obviously, these external segments could represent specialised dynamical elements rather than the controller structure if necessary. However, in developing PSICON it was clear that the structure clearly reflected the concept of the PSI simulator as the continuous process plant with the external segment as the controller algorithm. In this way, the controller structure is being prepared in a form suitable for implementation on a dedicated microcomputer. The implementation also uses the message transferring concept to incorporate interface considerations such as finite ADC and DAC accuracies. Further, since PSI runs on one graphics terminal giving chart-recorderlike responses similar to the measurement on plant, the external control segments can be used to represent the external environment including an operator console. In any case, the external segment has to run on a separate terminal, and in this way it can be run to appear like an operators' console with message information, or like a commissioning engineers' console give information on estimation parameters and performance criteria of the control etc. It should be emphasised that compilation and linking is only required for the external segments and is rapid in comparison with the equation-oriented multi-stage pre-processing. The external programmes are written entirely in Fortran, and so are transportable for implementation of the algorithms on site for the real process.

2. PSICON Facilities

The host simulator PSI which has been developed at Delft Technical University (van den Bosch, 1979) contains about 30 different block types. These include standard analogue simulator elements such as integrators, saturation devices, etc., and discrete elements such as zero-order hold, time delay and discrete PID controller. Certain logic functions are also included. Each block has a maximum number of 3 inputs and 3 parameters. The maximum number of blocks for a simulation is 150, with a maximum of 30 integrators. Other facilities available are algebraic loop provision, and a hill-climbing optimisation routine (Nelder and Mead, 1965) used for minimising cost functions set up in the simulation.

The PSI segment is entered and controlled by use of a simple command language. From a prepared simulation flow-diagram with named blocks, the structure is 'built' interactively. During simulation up to 4 selected blocks can be output to the screen either as time responses, or as phase-plane trajectories. Up to 4 outputs can also be stored for subsequent display on lineprinter output. A summary of the command language facilities is given in Table 1. 6 integration methods are available; two with variable step length, and 4 with fixed step length.

The external segment facility of PSICON provides the required additional flexibility to the powerful and fast block structure of the PSI simulator. The emphasis has been on the external segment being representative of an external control and identification algorithm ultimately aimed at microprocessor implementation. Thus although the whole simulation runs on a mainframe machine with strong integrity software and hardware, the studies are closely related to a prototype structure comprising a microcomputer and analogue simulator or a real process. To achieve the external controller concept, interaction between PSI and the control segment is via message transfer between the two separated tasks using the MTM multi-user operating system of the Perkin Elmer 3220 machine on which the language runs. The message transfer acts in a similar manner to computer/process interfaces. Thus, variables are transferred between the segments in integer format, and required accuracy of simulated ADC and DAC can easily be selected. The external controller segment is a Fortran routine, which can include subroutine calls as required. Only this segment requires recompilation when any algorithm changes are made. Since two tasks are established, these are managed from two terminals. The process simulation is run on a Tektronix 4010 graphics screen since it provides the continuous plotting of the process variables. The external segment and overall control of the simulation run-time is managed from another terminal. Normally this is an alpha-numeric terminal and can be used like an operator console. Alternatively, if graphical information is required from the control and identification algorithm, then a graphics screen can be used giving dual functions of operator control and 'commissioning' display. In this way the total simulation in certain aspects 'mimics' a real situation with one screen representing an instrumented plant, and the other the operating room giving overall control, tabulated data, message information such as alarm conditions, and commissioning data on software performance. The two examples which follow will illustrate some of these features.

3. Illustrative PSICON Simulations

3.1 Microprocessor Control of Muscle Relaxant Anaesthesia

In this project automatic control of levels of drug-induced paralysis in operating theatre is being achieved using an evoked EMG as the measurement of the degree of relaxation (Brown et al, 1980, Asbury et al, 1981).

PSICON has been used to set up PID controller parameters for humans and dogs and is currently being used to investigate alternative controller strategies. A block diagram of the system suitable for direct entry to the simulator is shown in Fig. 1. Using the optimisation routine in PSI with an integral error-squared criterion the best PID regulator time response is shown in Fig. 2. A sensitivity study of the system using known parameter fluctuations has shown that a fixed PID controller structure can give undesirable responses leading in some cases to an unstable system. For example the time response when one time constant is changed from 20 minutes to 10 minutes is shown in Fig. 3. Since the muscle relaxant dynamics contain a significant pure time-delay an alternative controller structure is the Smith predictor (Marshall, 1979). Using known parameters the optimised PID controller incorporated with a Smith predictor gives the improved response of Fig. 4. When the process and model parameters are mismatched an improved performance is still obtained, as indicated in Fig. 5 where one time constant has been changed from 20 minutes to 10 minutes as in Fig. 3. These initial studies were performed using PSI on its own, but both the PID controller and the Smith predictor algorithm have been implemented as external segments using PSICON. The external Smith predictor result is illustrated in Fig. 6, again showing superior performance to that of Fig. 2. The method of coding algorithms is important when using finite accuracy data based on ADC and DAC conversions. This is illustrated in Fig. 7

which is an external PID controller attached to the same muscle relaxant model. It can be seen that there is 'ripple' on the drug drive signal caused by the 10 bit quantisation via the ADC. A serious problem, however, is the large droop or offset in the output response, in spite of the fact that there was an integral term in the controller algorithm. The problem was a numerical one involving a coding which used 10 bit data accuracy on input and output and an algorithm which involved the differencing of small numbers. Problems of this nature are very easily revealed using PSICON with its external algorithms operating on finite accuracy data.

This anaesthetic example illustrates the ease with which a dynamic system including time-delay and heavy non-linearities can easily be simulated interactively using PSI alone. Simple controller structures can quickly be evaluated and then these and more complex algorithms can be transferred to externally coded segments suitable for microcomputer implementation. In this example on-line computer control of muscle relaxant administration has now been achieved using parameters obtained from the simulation studies referred to above. The implementation is via a Research Machine 380Z based on a Z80 processor with twin 5" discs, 56k RAM and CP/M operating system.

3.2 Self-tuning control of surface ship steering

There is currently much interest in adaptive control strategies based on the self-tuning principle. This principle maintains that simultaneous identification and control can converge to parameter settings for the controller which are near optimal. The major categories of interest are the 'implicit' self-tuner based on optimal control theory (Clarke, 1975) and the 'explicit' self-tuner based on classical servo-mechanism theory (Wellstead et al,1978). In the implicit method the controller parameters are estimated directly, while in the explicit method they are calcuated via estimates of the plant parameters.

Adaptive control schemes have been studied and implemented for course-keeping autopilots on surface ships (van Amerongen and ten Cate, 1975, Kallstrom et al 1977). The example shown here considers large-angle manoeuvring of a fast Naval frigate and the self-tuner is of the 'implicit' type (Mort and Linkens 1980). A block diagram of the continuous plant simulation is shown in Fig. 8. The yaw disturbance dynamics were modelled on the basis of known wave spectral behaviour. The ship dynamics are non-linear for large-angle maneouvres, and are modelled using a polynomial expansion (Bech and Smitt, 1969). The rudder servo and the rudder itself also contain non-linearities. In this example an extensive external segment is required to perform the self-tuning algorithm. Finite data conversion accuracy is included. Since convergence of controller parameters is crucial in this application, a second graphics screen was employed to give a continuous plot of the parameters as well as the time response of the ship dynamics.

Using a linear ship model the self-tuning time-response of Fig. 9 was obtained for 20^{o} rudder excursions, and an 8-bit conversion accuracy. The convergence of the controller parameters is shown in Fig. 10. When the data accuracy was changed to 12 bits the yaw time response was almost identical to Fig. 9, but the controller parameters showed greater fluctuations, particularly for the smaller parameters, as seen from Fig. 11. The steady state values, however, are very similar in both cases.

When non-linear ship dynamics, based on identification studies of a Naval frigate (Mort and Linkens 1981), were included, the self-tuner still performed satisfactorily. For large-angle manoeuvres (30^{o}) the time response of Fig. 12 was obtained. The equivalent response of the controller parameters is shown in Fig. 13, where it is seen that there are large initial fluctuations which subsequently die away leaving steady converged values. For comparison purposes the equivalent results for a PID controller optimised using the PSI hill-climber is shown in Fig. 14. It is evident that greater rudder saturation occurs for the PID controller than for the self-tuning auto-pilot.

4. Conclusions

The examples have illustrated the ease and flexibility with which PSICON can simulate continuous processors together with external control algorithms. These algorithms can range from simple PID regulators to complex self-adaptive structures. Initially, the interactive analogue-type simulator PSI can be used alone to obtain a 'feel' for the problem and eliminate flow-chart bugs. Its speed of re-running is impressive, being based on a pre-compiled block-oriented language.

Subsequent to initial simulation studies, the external segment can be enlarged progressively to include identification and control strategies. These algorithms are coded in Fortran, compiled quickly, and are in a form readily translatable into a form suitable for micro-computer implementation.

The two-terminal operation of the language is flexible, allowing the control segment to run as an operator-oriented simulation using an alpha-numerics screen, or as a development-engineer system using a second graphics terminal. In either case, plant behaviour is monitored directly as the simulation proceeds, and may be aborted if required.

5. Acknowledgements

The PSI stand-alone simulation software has been developed by the Control Systems group at Delft Technical University, Holland. The assistance of P.P.J. van den Bosch in transferring this software to the Perkins-Elmer 3220 is gratefully acknowledged. The anaesthetic muscle relaxant project is being carried out in collaboration with A.J. Asbury in the Department of Anaesthetics, Royal Hallamshire Hospital and B. H. Brown, Department of Medical Physics, Sheffield.

6. References

Brown, B.H., Asbury, J., Linkens, D.A., Perks, R. and Anthony, M.: (1980) 'Closed-loop control of muscle relaxation during surgery' Clin. Phy. Physiol. Meas., 1, p. 203-210.

Asbury, A.J., Henderson, P.D., Brown, B.H., Turner, D.J. and Linkens D.A. (1981): "Effect of diazepam on pancuronium-induced neuromuscular blockade maintained by a feedback system'. Br.J. Anaesth., 53, p. 859-863.

Bech, M. and Smitt, L.W. (1969): 'Analogue simulation of ship manoeuvres', Report Hy-14, Hydro and Aerodynamics Laboratory, Lyngby, Denmark.

Clarke, D.W. and Gawthrop, P.J. ; (1975), Self-tuning controller, Proc. IEE, 122, p. 929-934.

Kallstrom, C.G., Astrom, K.J., Thorell, N.E., Eriksson, J. and Sten, L. (1977): 'Adaptive autopilots for steering of large tankers', Report TFRT-3145, Dept. of Automatic Control, Lund Institute of Technology, Sweden.

Marshall, J.E. (1979) 'The control of time-delay systems', Peter Peregrinus, IEE Control Engineering Series.

Mort, N. and Linkens D.A. (1980) 'Self-tuning controllers for surface-ship course and track-keeping'. Symposium on Ship Steering Automatic Control, Genoa, June, p. 175-188.

Mort, N. and Linkens, D.A. (1981), 'Identification and Self-tuning control of linear and non-linear ships models', 6th Ship Control Systems Symposium, Ottawa, Oct.

Nelder, J.A. and Mead, R. (1965): 'A simplex method for function minimization', The Computer Journal, 7, p. 308-312.

Van Amerongen, J., and Udink Ten Cate., (1975): 'Model reference adaptive autopilots for ships', Automatica, 11, p. 441-449.

Van den Bosch, P.P.J., (1979): 'PSI - an extended, interactive block-oriented simulation program', IFAC Symposium on 'Computer-Aided Design of Control Systems', Zurich, p. 223-228.

Wellstead, P.E., Prager, P. and Zanker, P. (1978) 'Pole assignment self-tuning regulator', Proc. IEE, 126, p. 781-787.

TABLE 1 Command Language implemented in PSI

Define Simulation Data

B - go to block specification phase	DMi nam - define block to be stored in mem.i
P - go to parameter specification phase	DSi xi,xa - define scaling of memory i
F - go to function generator specification	DT dt - define integration interval dt
I - go to integration specification phase ph.	DP dp - define pring interval dp
O - go to output specification phase	Qi - select i-th integration method

K,G,T - select output representation; K = color TV, G = grafic to TT, T = table to TT

L - output is directed to line printer, else output is directed to teletype (TT)

Control Commands

R - run simulation, start for t = O	[CR] - carriage return- stop simulation
C - continue simulation	H - start optimization

Show Simulation Data

SM - show actual model	SS - show scaling and blocks stored in memories
SI - show actual integration data	SO - show values of the outputs of all blocks
SQ - show actual sorting sequence	SN - show names of blocks

Show Variables Stored In The Memories

XT ml - time response of memory ml (on TV)	XP - table of values of the variable stored in the memories on TT/line printer
XY ml,m2 - phase response of memories ml and m2	

Define Data For Optimization

HPi nam - i-th par. of nam becomes opt.par.	HD del - define initial step size del
HVi nam - i-th par. of nam is no longer - -	HE eps - define accuracy eps (stopcriterion)
HO nam - output of nam is cost function	HS - show all optimization data

Save And Read Models From Disk

MW nam - write model to disk	MD nam - delete model nam from disk
MR nam - read model from disk	MS - show all models stored on disk

Aabcdl234 - put text "abcdl234" above each output to TV, TT or line printer (max 40 characters)

N naml, nam2 - rename block naml into nam2

? - show all commands	?B - show description of all blocks

Figure Captions

Fig. 1: Flow diagram for PSI simulation of non-linear muscle-relaxant control system. Letters inside the blocks denote the type of block e.g. L = limiter. Names outside the blocks are the variables used in 'building' the simulation.

Fig. 2: Optimised PID step response for a non-linear muscle relaxant model using an integral of absolute error criterion.

Fig. 3: Step response of non-linear muscle relaxant model using same parameters as for Fig. 2 except that one time constant was changed from 20 minutes to 10 minutes.

Fig. 4: Optimised PID step response using a Smith predictor and the same model as that for Fig. 2.

Fig. 5: Step response using a Smith predictor for same parameters as for Fig. 3.

Fig. 6: Step response for non-linear muscle relaxant model using an external segment Smith predictor.

Fig. 7: An external segment PID controller with 10 bit accuracy ADC and DAC for non-linear relaxant model.

Fig. 8: Flow-chart for PSI simulation of surface ship manoeuvring dynamics and an external self-tuning autopilot.

Fig. 9: Self-tuning control response for a linear model of a frigate undergoing 20^{o} step rudder demands with 8 bit data conversion accuracy.

Fig. 10: Estimated controller parameters from the self-tuner result of Fig. 9.

Fig. 11: Estimated controller parameters using the same self-tuner as for Fig. 9 except for a data conversion accuracy of 12 bits.

Fig. 12: Self-tuning response of non-linear model of a frigate with sea-wave noise disturbance.

Fig. 13: Estimated controller parameters for self-tuning result of Fig. 12.

Fig. 14: An optimised PID controller response (based on an integral of absolute error) for the same model and noise as in Fig. 12. Note the increased rudder saturation.

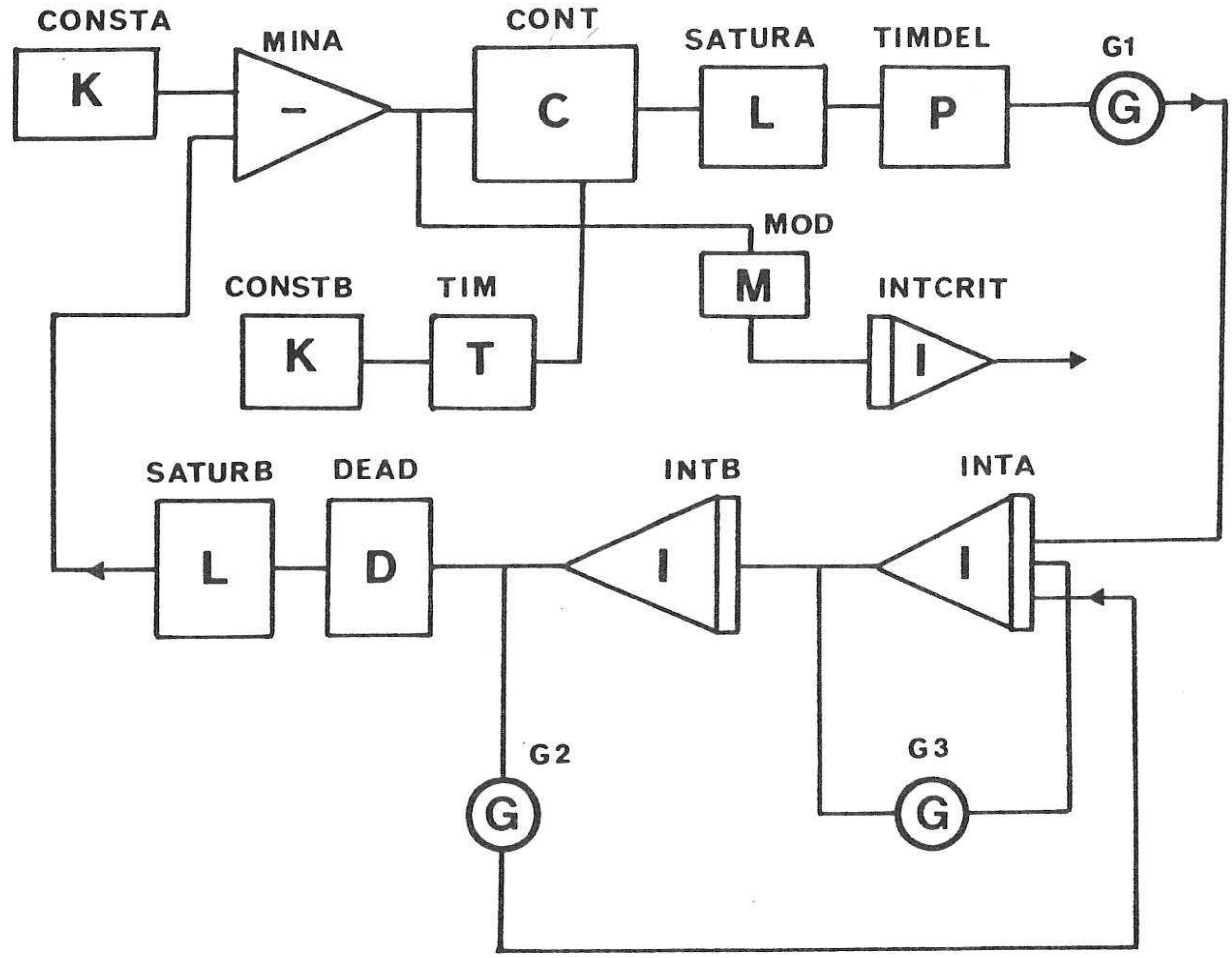

CONSTA
K
MINA
–
CONT
C
SATURA
L
TIMDEL
P
G1
G
MOD
M
CONSTB
K
TIM
T
INTCRIT
I
SATURB
L
DEAD
D
INTB
I
INTA
I
G2
G
G3
G

Fig 1

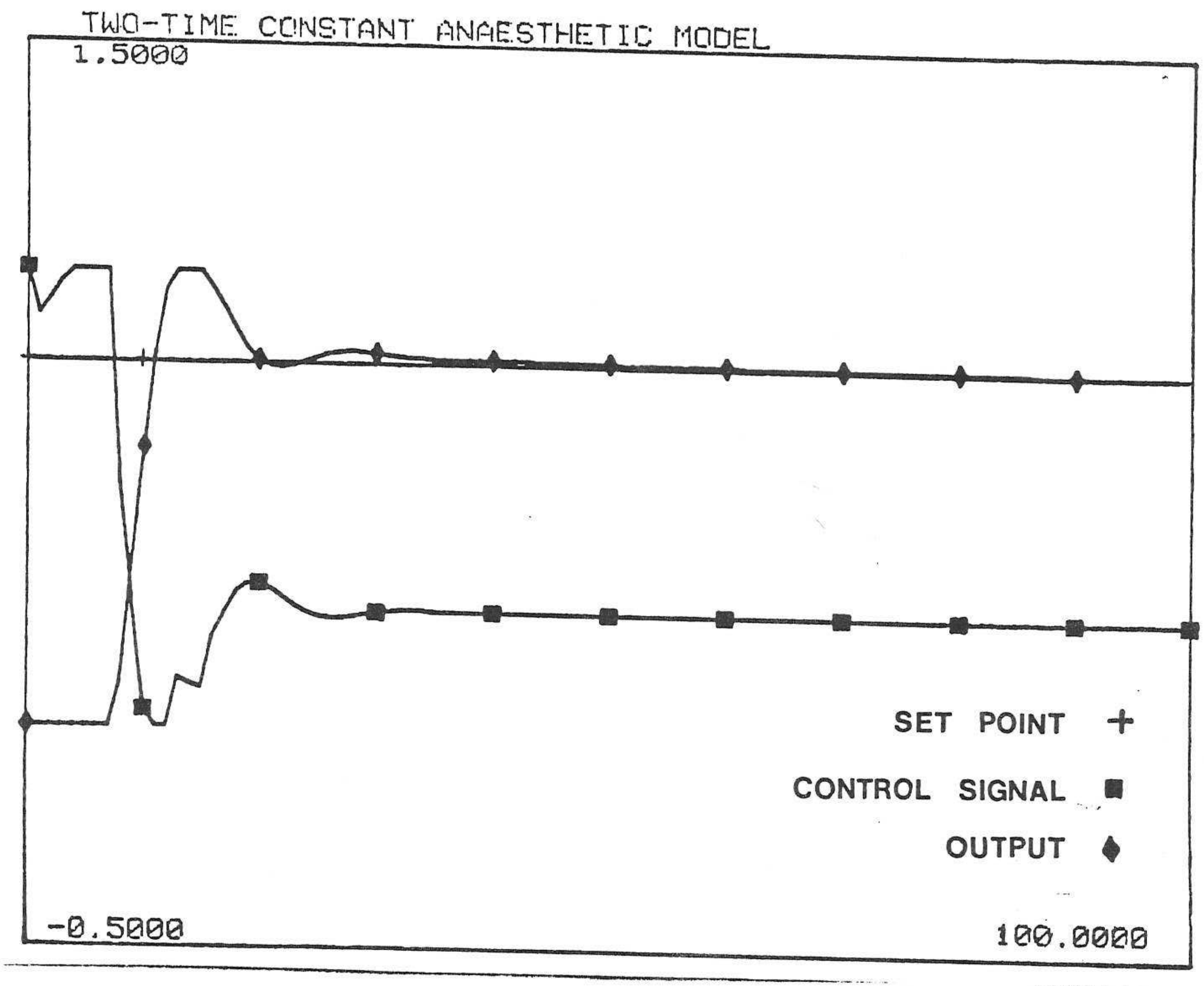
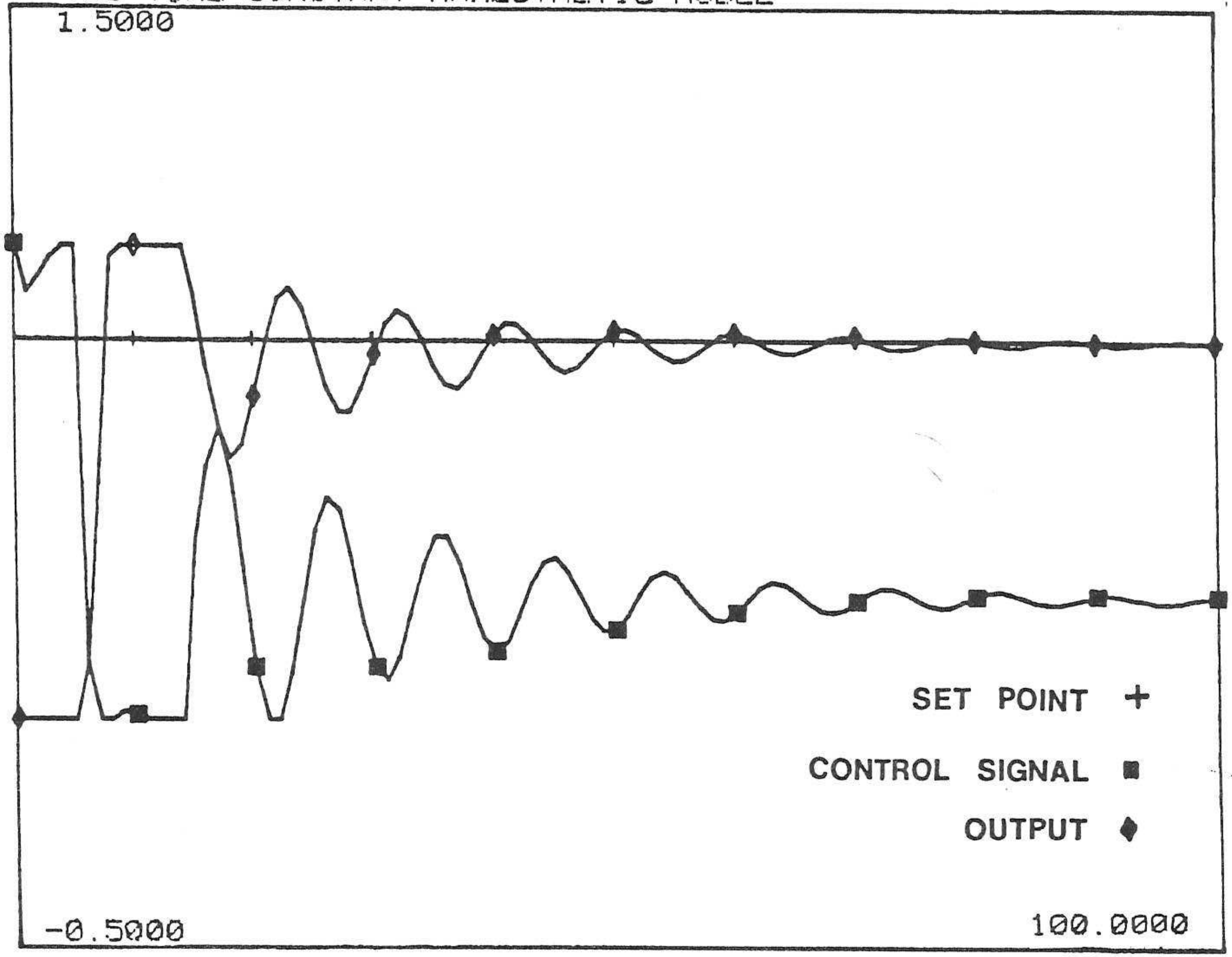
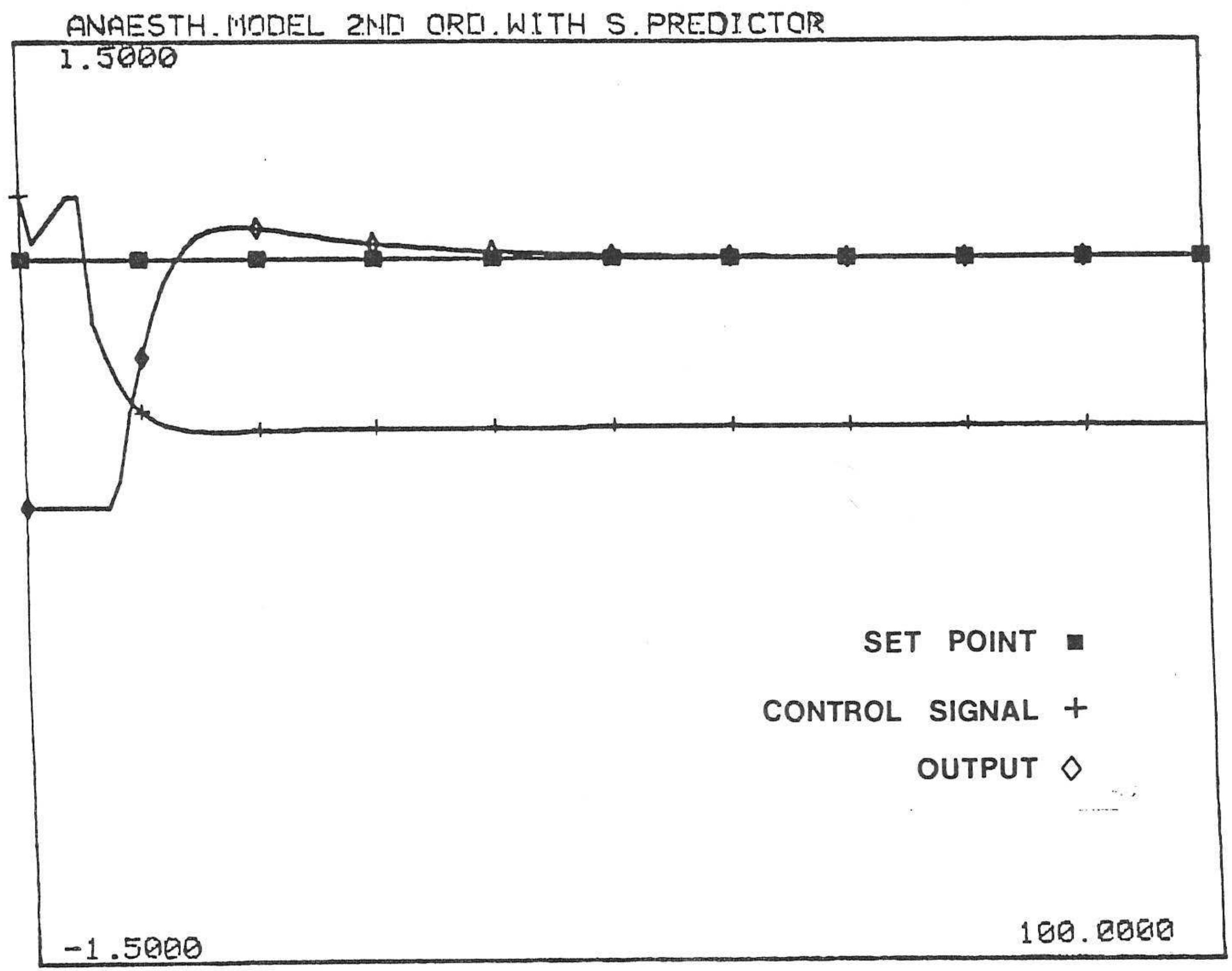

TWO-TIME CONSTANT ANAESTHETIC MODEL
1.5000
-0.5000
100.0000
SET POINT
CONTROL SIGNAL
OUTPUT

Fig 1

TWO-TIME CONSTANT ANAESTHETIC MODEL

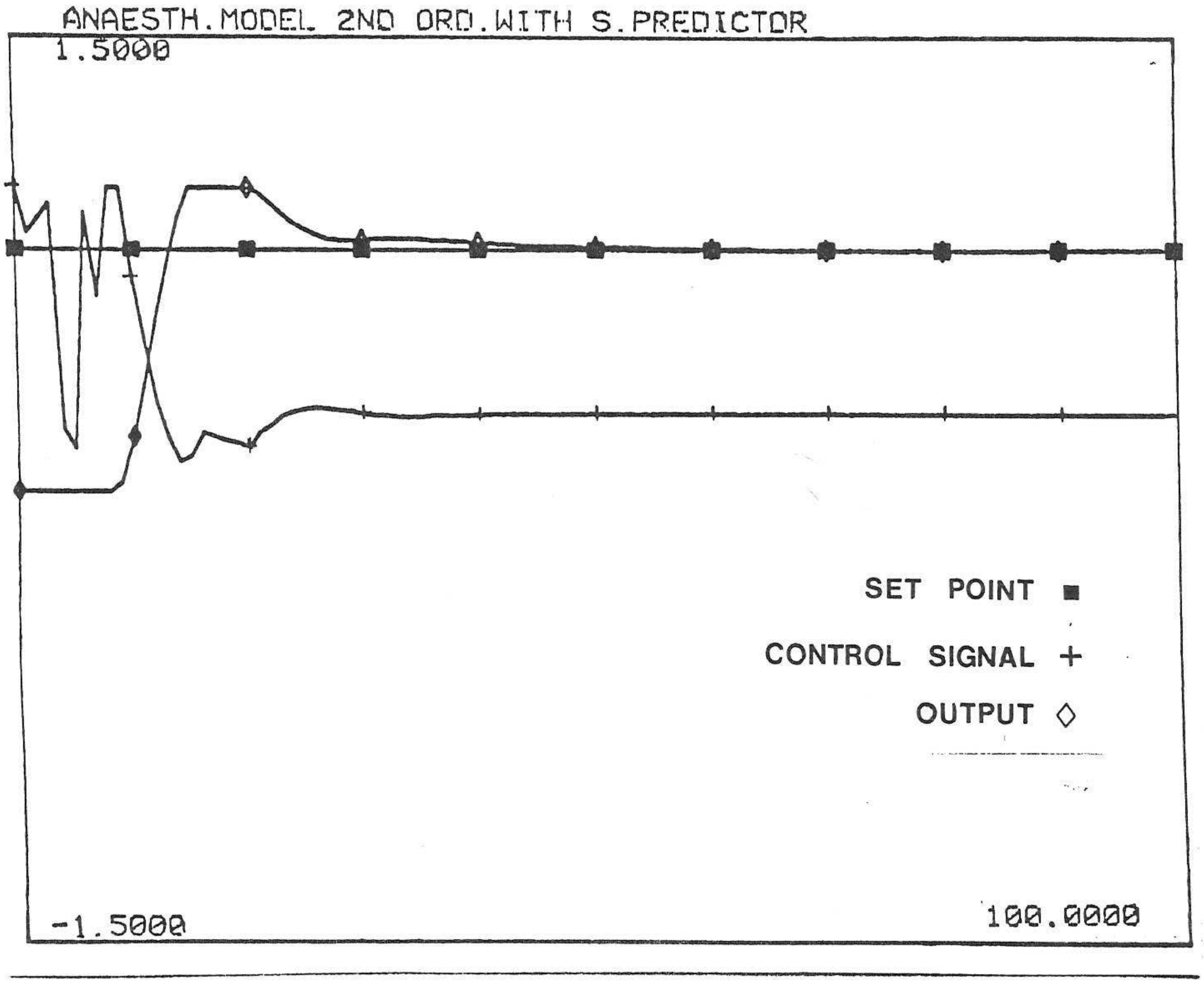


7ia


ANAESTH.MODEL 2ND ORD.WITH S.PREDICTOR
1.5000
SET POINT ■
CONTROL SIGNAL +
OUTPUT ◇
-1.5000
100.0000

4

Fig


ANAESTH.MODEL 2ND ORD.WITH S.PREDICTOR
1.5000
-1.5000
100.0000
SET POINT ■
CONTROL SIGNAL +
OUTPUT ◇

Fig

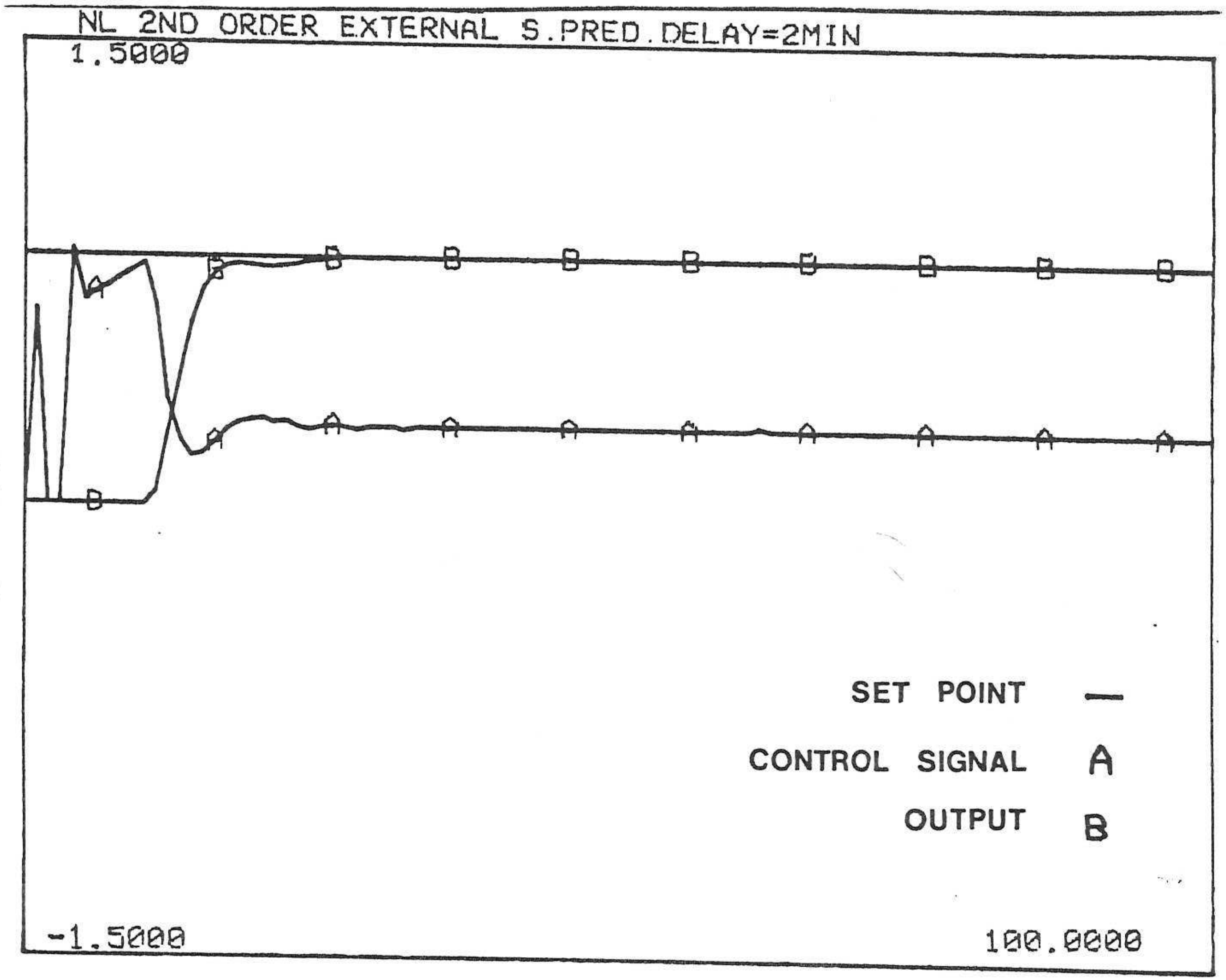

NL 2ND ORDER EXTERNAL S.PRED.DELAY=2MIN
1.5000
SET POINT —
CONTROL SIGNAL A
OUTPUT B
-1.5000
100.0000

FIG. 6

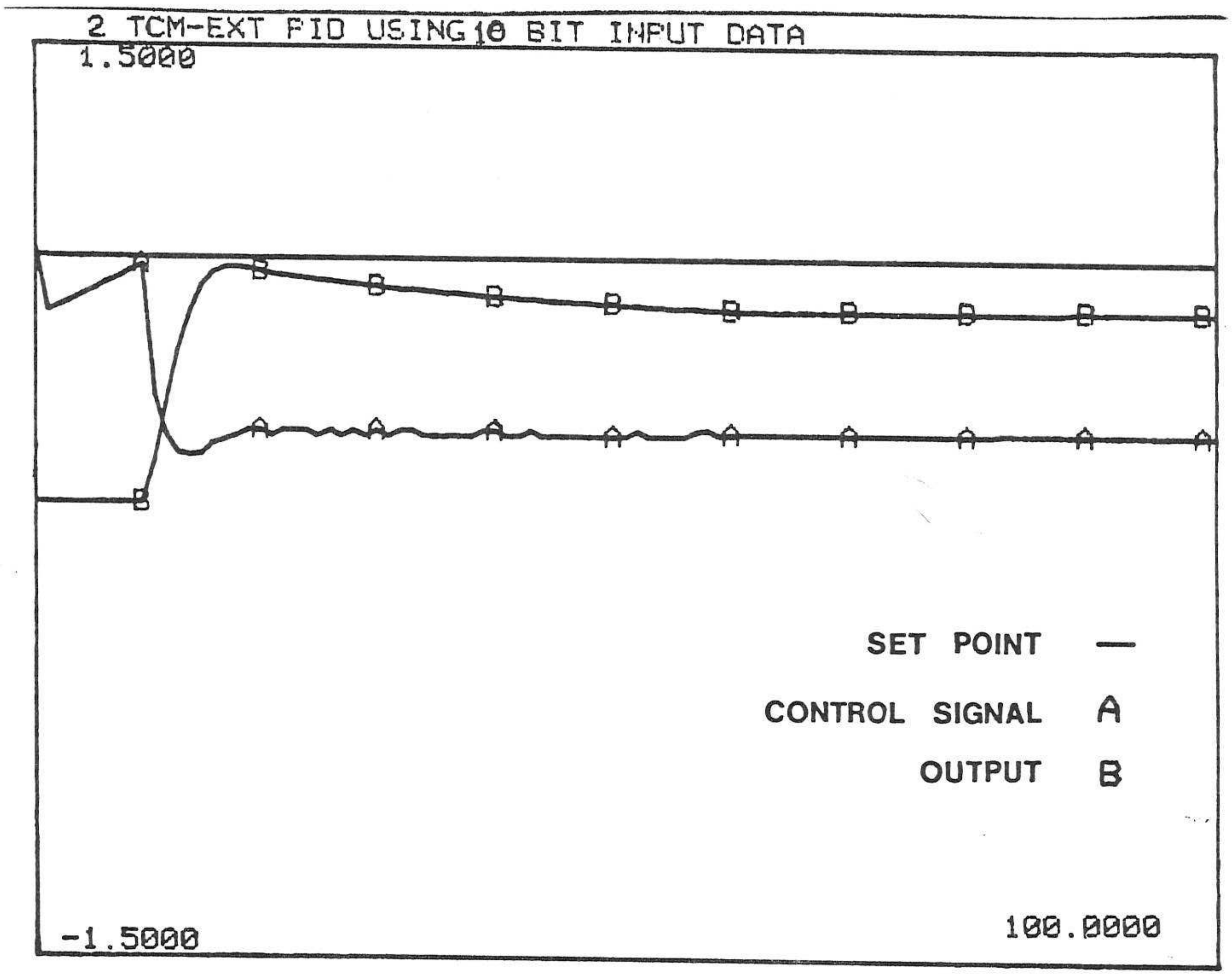

2 TCM-EXT PID USING 10 BIT INPUT DATA
1.5000
SET POINT —
CONTROL SIGNAL A
OUTPUT B
-1.5000
100.0000

FIG. 7

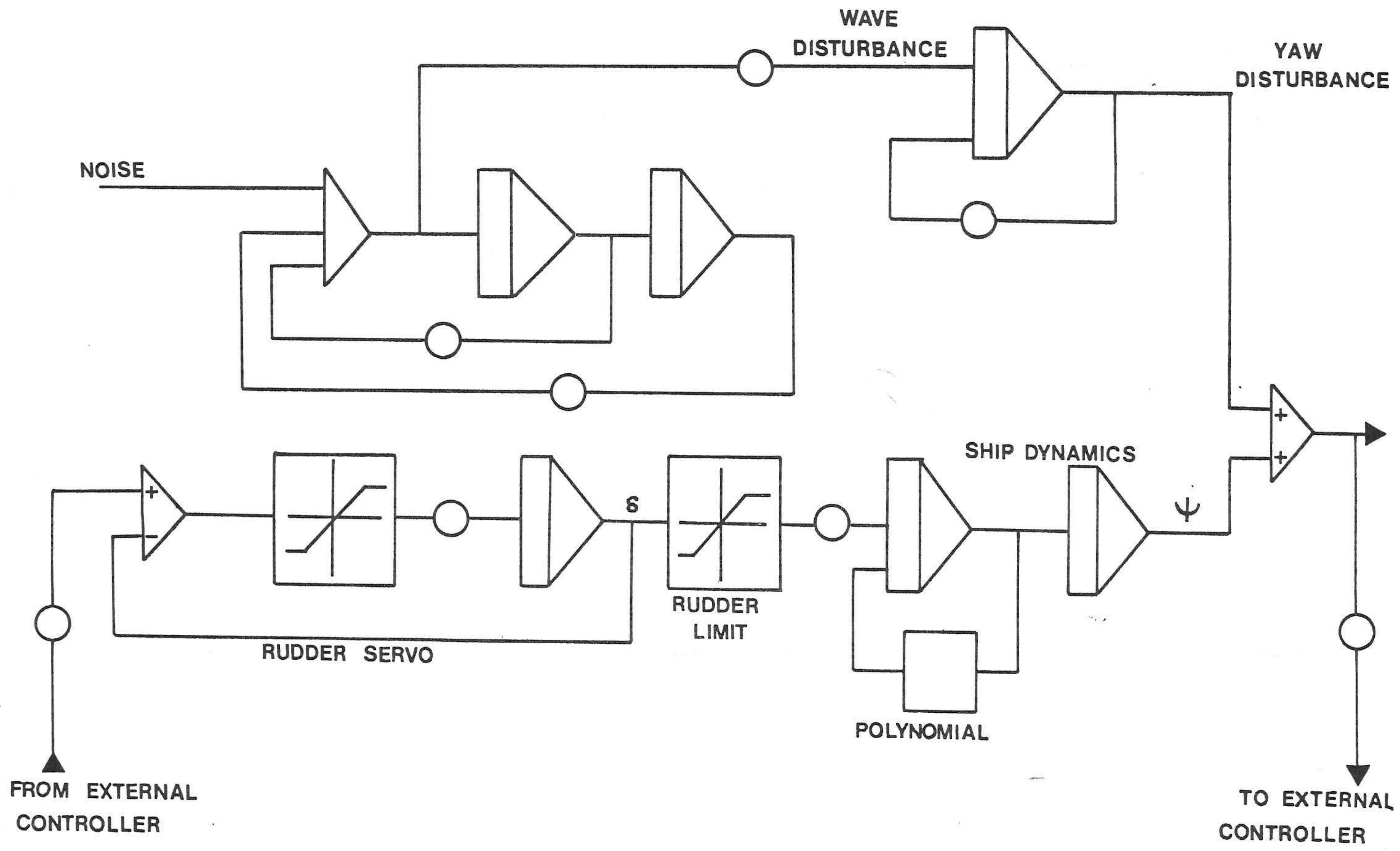

WAVE
DISTURBANCE
YAW
DISTURBANCE
NOISE
SHIP DYNAMICS
δ
ψ
+
+
+
-
RUDDER
LIMIT
RUDDER SERVO
POLYNOMIAL
FROM EXTERNAL
CONTROLLER
TO EXTERNAL
CONTROLLER

FIG. 8.

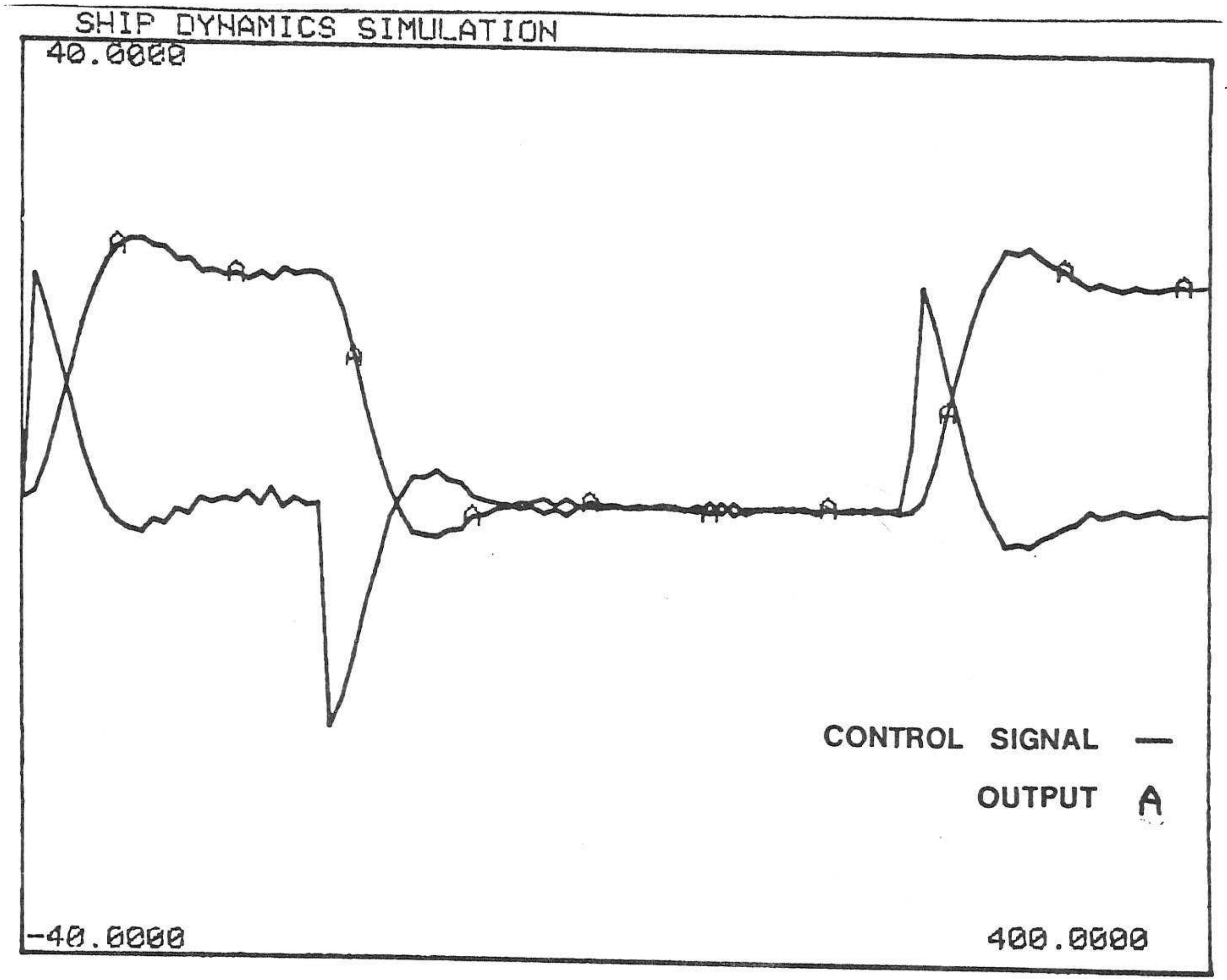

SHIP DYNAMICS SIMULATION
40.0000
-40.0000
400.0000
CONTROL SIGNAL —
OUTPUT A

Fig. 9

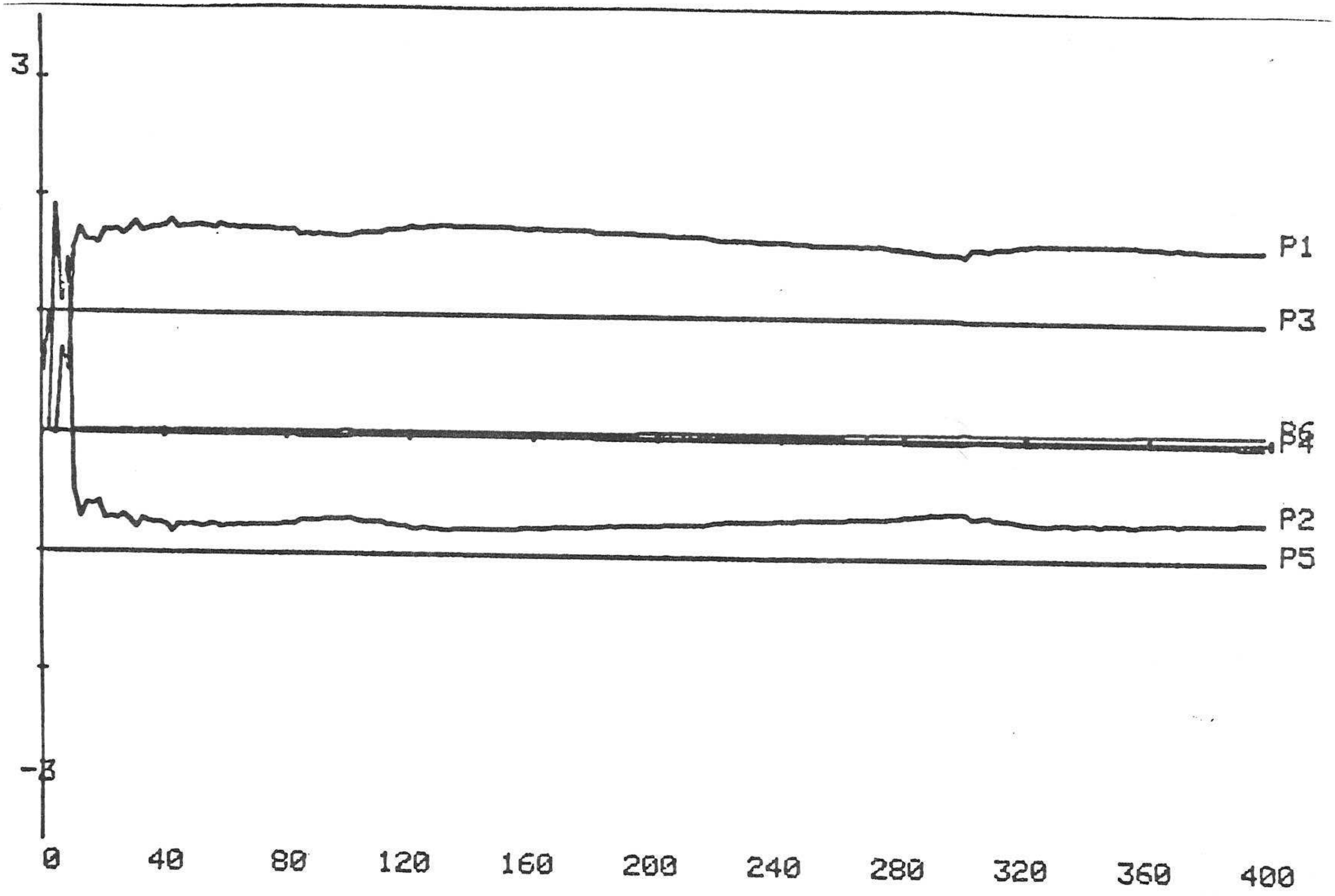


Fig 10

11

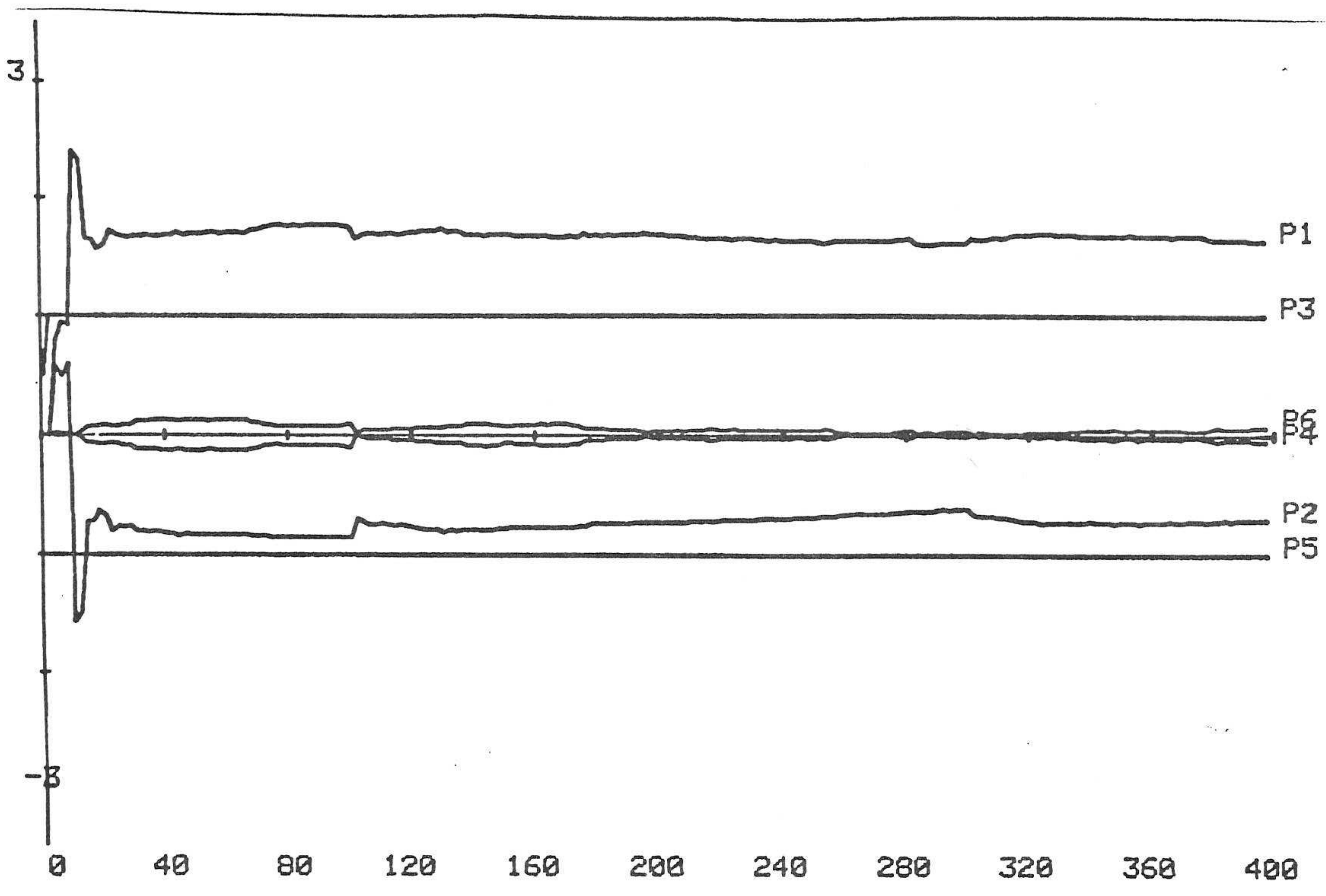


Fig. 11

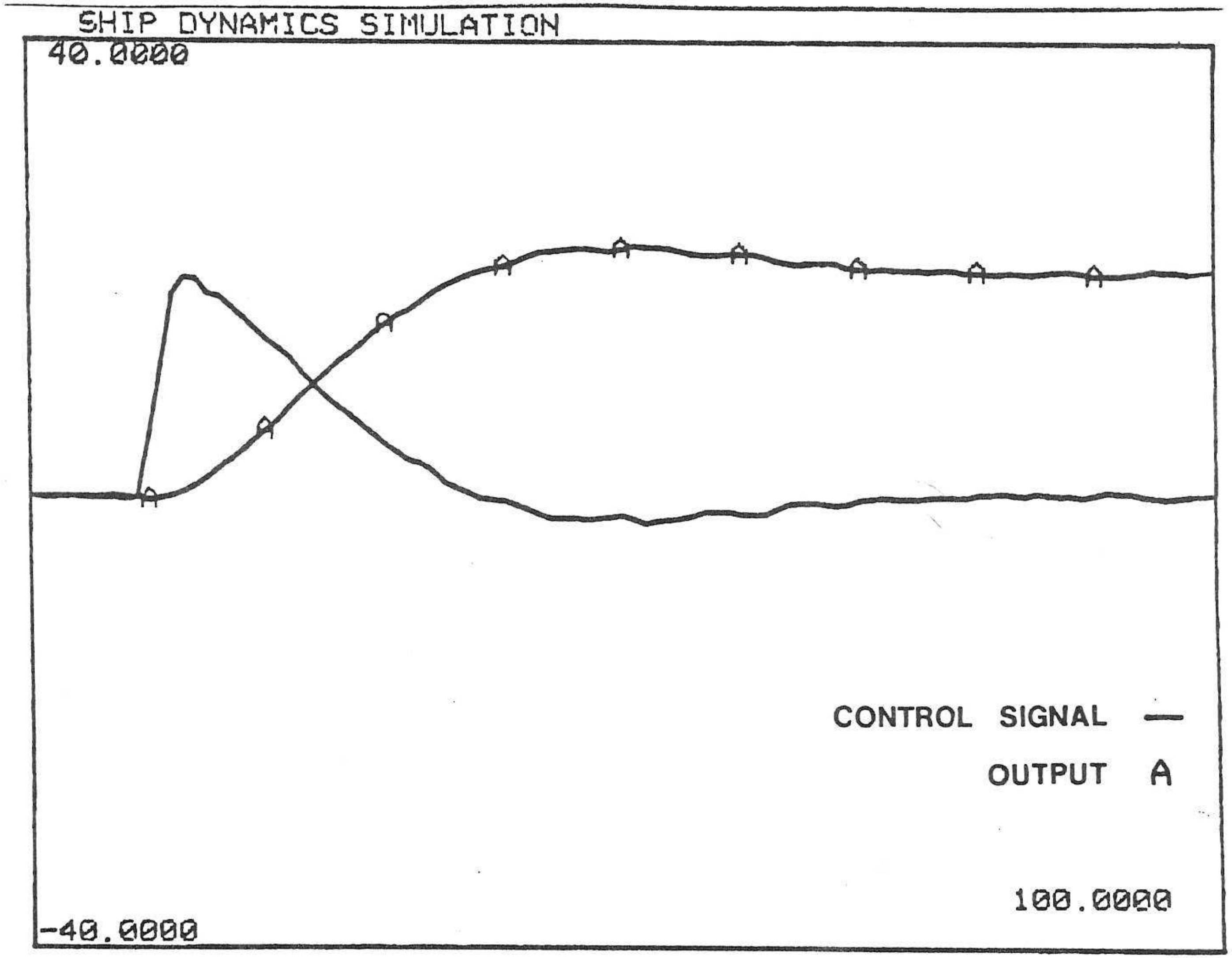

SHIP DYNAMICS SIMULATION
40.0000
CONTROL SIGNAL —
OUTPUT A
100.0000
-40.0000

Fig 12

13

Fig 13