SOFTWARE SIMULATIONS OF ON-LINE ANALYTICAL SYSTEMS

Dr. V. Kratky
Photogrammetric Research
National Research Council of Canada
Ottawa, Ontario, CANADA, K1A 0R6
Commission II

ABSTRACT

The software for on-line analytical systems is generally not portable from system to system due to many differences in their hardware and interface structure. The efforts in developing more universal software would greatly benefit if the programming and testing work could be carried out independently of the system hardware. An analysis demonstrates how to simulate the analytical plotter and interface functions by software, using a suitable interactive terminal. Two examples of experiments document a successful simulation of an analytical plotter using either a computer graphics terminal via a lightpen interaction, or a regular computer video terminal with interaction achieved in a split screen mode by a moving cursor.

INTRODUCTION

Most analytical plotters are controlled with a minicomputer support, sometimes in combination with an additional microprocessor power programmed to perform some routine, real-time functions. The program for one system is not compatible with another one even when the same minicomputer is used. Software is generally not portable due to many differences in the system architecture, in the effective control of individual components and in the design of the interface. As a consequence, any on-line software development is useful only for a particular instrument which must be available for testing. Efforts aimed at the higher standardization of on-line analytical systems and the development of more universal software are rather restricted.

The situation would improve if some programming and testing work could be carried out without the presence of the system hardware. It is feasible that the hardware functions be computer simulated and programs run with a simulated interaction just through the computer with no plotter hardware involved. This approach may stimulate more independent software development and also provide an excellent tool for training and demonstrations.

Obviously, simulations do not generate any meaningful measurements, as no photographs are involved, and thus the accuracy and reliability of real systems cannot be evaluated. However, most other practical aspects, such as programming structures, the organization of interaction, the efficiency of computations etc. can not only be tested, but also refined through simulations.

FUNCTION OF ON-LINE SYSTEMS

Before starting to analyze how to proceed with hardware simulations it is useful to review the basic functions of on-line systems, with reference to more detail in (Jaksic, 1973, 1980). Generally, on-line digital photogrammetric systems are photograph measuring devices (comparators, plotters) interfaced with a computer which can process measured information on-line with its collection. Of interest here are only closed-loop systems

(Figure 1) which not only instantly process the information, but also use it in a feedback mode to monitor and control the operation. The main element of this control is a continuous, computer controlled, real-time positioning of images. The interface between the computer and the photo-carrying hardware allows for a two-way communication. On-line photogrammetric systems of this type are called analytical plotters.

Existing analytical plotters differ widely in their design, computer support and interface implementation (PERS, 1982). However, there is not a significant difference in their basic functions and the operator interacts with all these systems in a very similar manner. This unifying factor of the operator-system interaction and control structure makes it possible to analyze the potential of computer simulations of analytical plotter operations in a relatively general approach. In order to do that one should first look at the overall structure of data processing control in individual systems. From the viewpoint of the processing control there are essentially two types of systems, with central processing and distributed processing.

Central Processing

On-line systems with central processing rely on a single computer in a configuration corresponding to a general sketch of the closed-loop approach in Figure 1. More detailed block diagram in Figure 2 replaces the simple block [PHOTOS] by [ANALYTICAL PLOTTER] showing all its individual components important for both the internal and external control structure. The operator supplies manual input to coordinate registers, some of which are then serviced by [SERVO CONTROL] and this module in turn positions photo stages with respect to [OPTICS], thus providing an optical feedback to the operator, as needed for his control of measurements. The inner servo-loop which involves [REGISTERS], [SERVO CONTROL], [PHOTO STAGES], usually operates independently of the computer. Additional input from the operator comes through console switches and affects the function of registers. Display counters can be available on the console to provide a readout from some registers. The computer communicates directly with the analytical plotter through registers and switches. Coordinate values are transferred to and from registers, while switches are computer interrogated to affect the control of operations. Both functions are serviced by special interface subroutines. Finally, the operator communicates with the computer directly, through a standard video terminal. The above description is conceptual and if reflects well the function of some analytical plotters, such as NRC Anaplot, Zeiss C-100, Wild AC 1. The number and function of console switches differ from case to case.

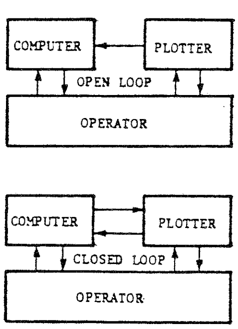

Fig. 1 On-line control

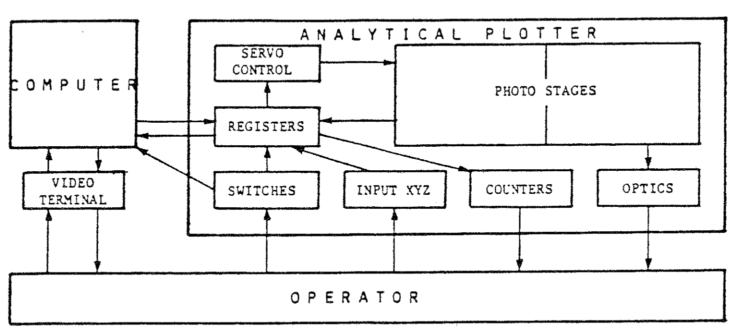

Fig. 2 System with central processing

For instance, some of the switch functions can be controlled from the video terminal and this terminal can also display the coordinates instead of using console counters.

Some advanced systems have additional computer control of the observed optical image, affecting the magnification and rotation of viewing fields. In the context of this paper this optical control is irrelevant and is disregarded in Figure 2.

Distributed Processing

The concept of distributed data processing is based on the use of a supporting computing power of microprocessors which in effect relieves the function of the computer. The processor is preprogrammed to perform certain standard or utility functions via firmware. It can then be equipped merely with 'read only' memory, while variable parameters are supplied from the computer. Figure 3 renders an example of a two-level distribution, as typical for the HAI US-2 analytical plotter. The processor eliminates the real-time positioning function from the computer control. Coordinate registers are attached to the processor which then receives the X, Y, Z input from the operator. The processor assumes full responsibility of real-time position calculations and servo control. The switch functions are also directed through the processor. Another example in Figure 4 illustrates a three-level

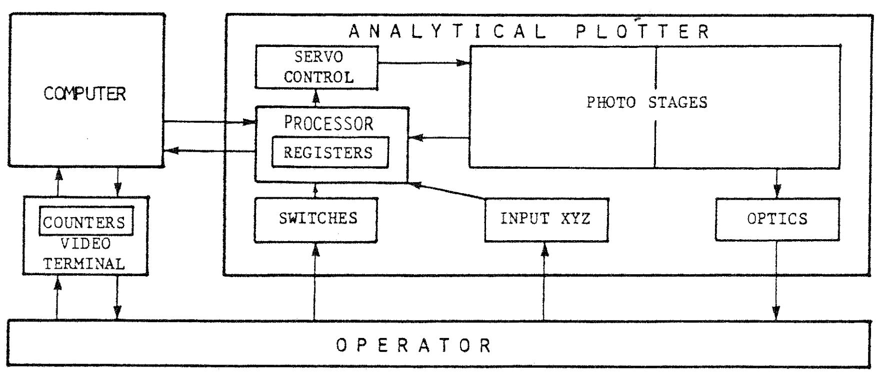

Fig. 3 Two-level distributed processing

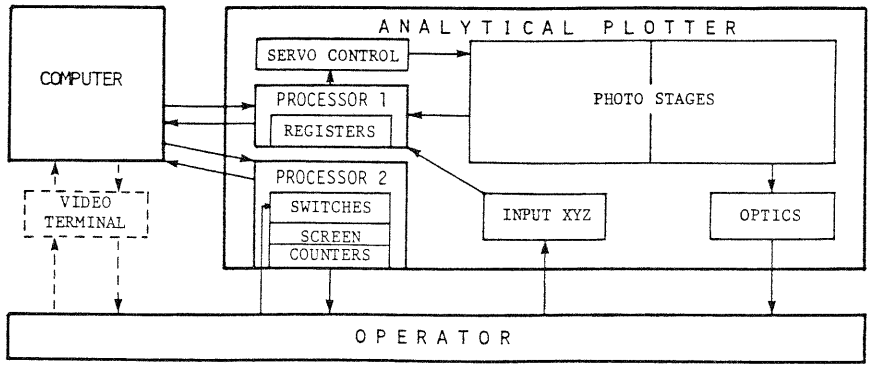

Fig. 4 Three-level distributed processing

distribution of processing, as implemented in the Kern DSR1. Processor 1 takes care of the real-time positioning functions, while the front end processor 2 communicates with the operator. The operator receives information from a flat video screen and coordinate counters located at the plotter console, and his control input is entered from special switches and a numeric keypad. A video terminal providing a direct link to the main micro- or minicomputer is optional.

Software Control of Functions

Most analytical plotter functions are controlled by software, however, they have to be initiated by the operator. Operator activated switches are then detected and interpreted to interrupt/resume or to abort an operation, to start an action and to select a mode of current operations. The program must anticipate these signals, otherwise the operator's requests would go unnoticed. This is done by the interrogation of relevant switch status words during all indefinite measuring cycles, expected to be terminated or modified by the operator.

OPERATOR-SYSTEM INTERACTION

The operator interacts with any on-line photogrammetric system in two ways: through his video terminal link to the computer and through the input/output options of the analytical plotter. The computer directed interaction has several components:

- solicited input based on a dialog; it is conducted either as a prompt/answer conversation or is menu driven, offering a multiple choice at several recursive levels;
- unsolicited input needed to interrupt/resume an operation, to break for another independent operation in a multitask environment and to abort the task;
- computer output of intermediate or final results;
- continuous display output presented as a time scheduled screen output.

As shown in Figure 2, the operator-plotter interaction is based on two input channels and two output channels. The operator receives a continuous, dynamic feedback on the progress of the operation from the optical system. He can also monitor current coordinates or solution parameters in display counters available on the system console or through the video screen. Based on this feedback the operator manually controls his X, Y, Z input transmitted and processed in real time. In addition, there is a secondary, occasional interaction through the switches. In most cases the switch signals affect the program decision making, however, some of the switches control and modify the hardware performance, e.g., the information flow between registers.

COMPUTER SIMULATION OF HARDWARE FUNCTIONS

Any simulation must model the corresponding physical reality and accurately reflect its specific characteristics. For reasons of space limitation the following analysis will be less general and will address a system with the central processing of data.

General Approach

The operator-computer dialog and interaction is controlled by existing analytical plotter software which is fully preserved in simulations, except for a few routines servicing the computer-plotter interface, as explained below.

The internal hardware structure must be represented by suitable program modules which simulate the individual hardware functions. All physical registers and switches are replaced by suitable memory locations functioning as pseudoregisters and pseudoswitches, and all original hardware functions are programmed as equivalent manipulations of data among their pseudo-counterparts. In other words, the analytical plotter is now represented by a new subroutine, PLOTTER, which is called from the main program in all its parts normally operating on interface registers and switches. Except for photo stages which are represented in the simulation just by currently considered photo coordinates, all remaining six plotter subblocks of Figure 2 must have equivalent sections in the PLOTTER subroutine. They will be individually discussed below.

The computer maintains a link with the analytical plotter through registers and switches. For registers it is a two-way communication supporting both read and write functions, while switches are only signal transmitters. All register- and switch servicing subroutines of the original program must be modified to substitute newly defined pseudoregisters and pseudoswitches for the real ones. This usually applies only to a few interface oriented subroutines of the main software and is easy to accomplish.

The operator's link with the analytical plotter is more difficult to simulate and cannot be represented only by software. The simulated communication must be supported by some standard, general-purpose interactive medium connected to the computer as a substitute for the analytical plotter. It may be an interactive graphics terminal, a programmable tablet attached to the computer or just the available video terminal alone, if it is of a suitable type. With reference to Figure 2, the input to [ANALYTICAL PLOTTER] block through [SWITCHES], [INPUT XYZ] and output from [COUNTERS], [OPTICS] is simulated by interacting with the graphical representation of these functions, with the use of a light pen, stylus or screen cursor.

Register Control

All coordinates registers can receive, capture and transmit numerical information and this is done in a logical system of communication lines defined by the functions of any particular analytical plotter. In the following we will examine register functions of the NRC Anaplot. Analogy can be used to analyze other systems.

For a real-time operation the most important are stage registers involved in the servo loop control. Each stage coordinate has two registers, demand register D with input coming either from the operator or from the computer, and status register S reflecting the current position of photo stages from their encoders. In order to close a servo positioning loop stages are automatically driven to eliminate any difference between D and S which eventually assume the same contents, except for some control fluctuation of the lowest bits.

In the manual input of X, Y the transfer of increments from handwheels to D-registers is directed under control from both the computer and operator. With the computer-set register switch M (for manual status) the handwheels increment both model and stage registers. When M is computer cleared the manual input goes only to model registers and stages receive corresponding computed values from the computer. The operator also affects the X, Y input by footswitch P (for parallax status). When P is activated the input is transferred only to D-registers of one of the stages currently defined by the state of another manually operated L/R (left/right) switch and bypasses model registers. The Z-wheel generated input is always simply directed to the Z-register.

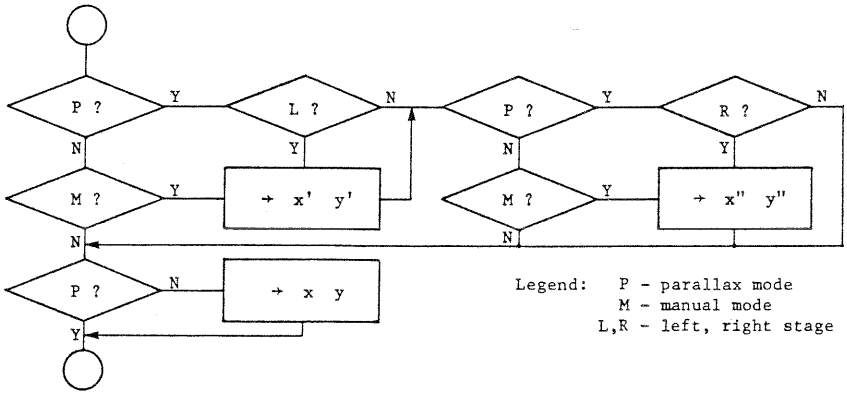

Fig. 5 Register incrementation from handwheel input in NRC Anaplot

The above hardware logic must be fully reflected by the corresponding program logic in simulations. Hardware generated pulses are simulated by coordinate increments from the 'INPUT XYZ' module, directed to and accumulated in pseudoregisters, as controlled by memory located pseudoswitches M, P, L, R. The flowchart in Figure 5 reflects the above functions.

Stage Servo Control

The hardware implementing the control of the servo loop operates on D- and S-photo registers. Depending on the type of the analytical plotter the servo loop is closed gradually by generating suitable voltages for servo motors, while applying some mechanical acceleration and velocity control to achieve a smooth and vibration free homing-in of photo stages. A program simulating this process can exactly replicate the closing function by iteratively adjusting the content of S-pseudoregisters to that of D-pseudoregisters. However, because this program is alternately run for all four photo coordinates and is dependent on a higher programming loop containing the interface servicing routine from which it is called, the simulated homing-in may be time consuming for longer distances to be travelled. It is then appropriate to simplify the simulation process in order to achieve a response time which is commensurate with the timing of the original hardware. In the extreme case the S-pseudoregisters can be directly assigned the current value of D-pseudoregisters to close the loop immediately.

Hardware Control

Switch Control. Hardware switches of an analytical plotter are connected through interface in such a way that their manual activation immediately results in setting or clearing flag bits of memory located switch status words. The switch control part of the PLOTTER routine takes care of identifying the position of any corresponding pseudoswitch in the graphical representation of the switch control on the interactive screen. Once such a position is selected and identified, e.g., with the use of an interactive light pen, the corresponding flag bit setting and clearing is accomplished by the PLOTTER routine. The main software then processes the information so encoded in the switch status words in the same way as if the real switches were operated on in the analytical plotter.

Coordinate Input Control. Manual input X, Y, Z to analytical plotters can be accomplished with various control elements, such as handwheels, footdisk, joystick, trackball, handdisk, cursor etc. Regardless of its type, each of these input elements generates a train of pulses which are incrementing appropriate model registers or photo D-registers. In the simulation approach the best way is to display a graphical rectangular array representing the range of a joystick operation for the X, Y input. With the origin in the centre all other discrete interaction-sensitive positions in the array are assigned the meaning of suitably increasing positive/negative increments in X and Y direction. Similarly, a one-dimensional array of positions displays a suitable range of Z-increments. During the simulation run the operator activates a suitable position in the arrays, as if he would operate a joystick. The input control part of the PLOTTER routine will use the specified values to increment the pseudoregisters, as described above. The incrementation of pseudoregisters will continue in each calling cycle to PLOTTER until the input is deactivated by selecting a zero-increment value positioned in the centre of the pseudo-joystick arrays. The currently active position within the arrays is always graphically marked to allow the operator to monitor the input process.

Hardware Output

Digital Displays and Counters. Console-mounted digital coordinate and parameter displays must be simulated on the interactive screen. For analytical plotters with a time scheduled continuous display outputs on the video terminal the simulation is not needed. In the opposite instance, the interactive terminal will display appropriate numerical values in suitably positioned annotated counters. The COUNTERS part of the PLOTTER routine takes care of the scaled transfer of the contents of individual pseudoregisters.

Optics. The main feedback for the operator control of a regular analytical plotter operation comes from the optical system. Even though a simulation of a measurement process controlled through the optical system is not meaningful, it is still important to simulate at least a limited joystick-controlled displacement of measuring marks in fictitious optical fields. They are displayed on the interactive screen as windows with a centrally located fixed measuring mark. Small changes of the pseudoinput X, Y, Z are reflected in the optical windows as corresponding displacements of optical pseudodetail represented on the screen by a suitable symbol. The displacement is controlled by the OPTICS part of the PLOTTER routine. By using this option one can simulate the introduction or elimination of small errors in recorded pseudomeasurements and effect a more realistic computation of not-only-zero elements of transformations and orientations involved in the fictitious run of the analytical procedure.

Control of Simulation

In on-line operations any individual operator-plotter interaction, with the exception of unscheduled breaks and interrupts, can be effected only at times during which the program anticipates operator's input by interrogating the state of switches and contents of registers. The program control is running in indefinite cycles temporarily suspended or eventually terminated when operator's commands are detected. These cycles are usually programmed to collect and process measurements either in the stereocomparator mode or in the real-time, computer controlled mode of stage positioning. It is obvious that these program sections already contain calls to interface servicing routines to read or write values from or to registers. Let the names of these existing routines be GETREG, PUTREG.

To make sure that the simulation control is complete and all analytical plotter operations are properly represented in simulations, a call to the PLOTTER subroutine must be placed from inside of existing GETREG and PUTREG routines. This approach will also accommodate a proper simulation display of any positioning operations needed, e.g., to drive to loading or centre positions of stages, to slew in specified point patterns and restore already recorded positions. The GETREG and PUTREG modules are part of the existing interface library which has to be modified anyway, to adapt all its functions due to the conversion from the real environment to its fictitious substitute. A great advantage of this approach is that the bulk of the analytical plotter software is not affected and can be used in simulations unchanged.

The PLOTTER subroutine must, of course, also take care of the proper integration of the new interactive medium and this depends on the hardware used. To operate from an independent graphics terminal or tablet involves additional software, usually a library of supporting subroutines for that particular peripheral. In operating from the same video terminal one must implement, among others, an additional split screen manipulation of data to ensure a clear separation of the regular monitoring of analytical plotter operations from the interactive displays of the simulated functions on the same terminal.

IMPLEMENTATION

Experiments have been conducted to simulate the functions of the NRC Anaplot and to test the feasibility of analytical plotter simulations in two different configurations: with the use of a graphics terminal DEC GT-40 and with the use of a standard video terminal DEC VT-100. The host computer to the Anaplot is DEC PDP11/45.

Graphics Terminal

The GT-40 terminal allows one to interact with its screen information via a light-sensitive pen. Any of the displayed lines, figures or letters can be defined sensitive to the pen close pointing. Thus, the position of a 'hit' is identified and its significance or meaning interpreted by the program as an activation of a pseudoswitch or input from a pseudo-joystick of the simulated system. Figure 6 illustrates the screen configuration. The upper part represents output information from the fictitious plotter: two circular optical fields with a measuring dot in the centre, displaced crosses representing the optical details, a square frame to show the current positions within the area of both photographs (L, R) and corresponding photo-coordinate counters. Displayed in the lower part are Anaplot control switches and two joystick arrays with coordinate counters to control operator's input in planimetry and elevations. Any of the boxed switch letters or numbers is sensitive to the light pen and, when touched, they start blinking to indicate activation. With the light pen hit of the joystick areas a small circular shaped cursor appears on the screen and can be moved around to select the speed of X, Y, Z incrementing as in the operation with a pressure-sensitive joystick.

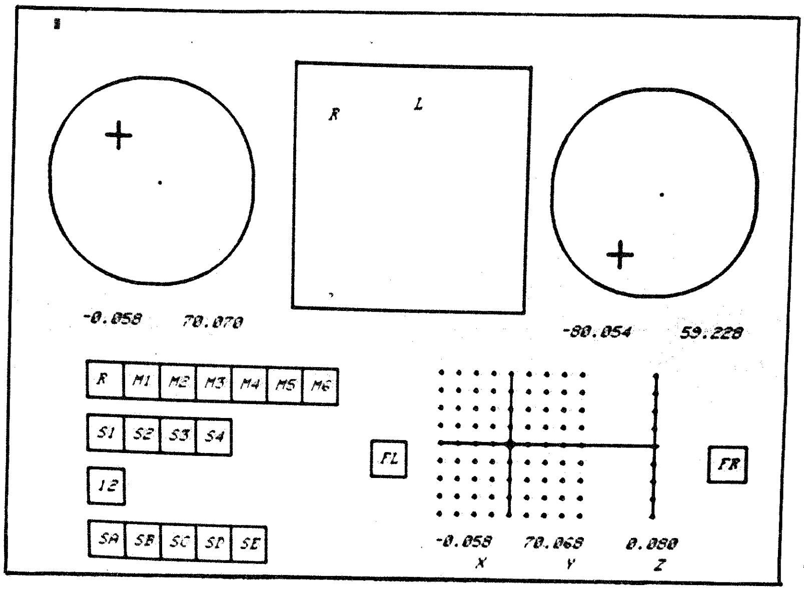

Fig. 6 Anaplot simulation by GT-40

Standard Video Terminal

Simulations run on standard computer-terminal configurations, with no additional hardware involved, are more practical and useful, therefore, most experiments were carried out with the use of the VT-100 video terminal. The screen is horizontally split; the upper part is stable and arranged as shown in Figure 7, to represent all Anaplot controls and outputs. The lower part is then used for the regular computer-operator dialog and display of results in a scrolling mode. VT-100 is not a graphics terminal and its screen resolution is defined only by the size and spacing of displayed characters. However, some of its programmable functions have an interaction potential and, with certain limitations, the terminal can be used for on-line simulations. The terminal functions and modes of display are controlled and can be programmed by command sequences of characters entered from the keyboard or transmitted from the computer. The sequences must start with a special keyboard key, marked ESC, or with octal 33 transmitted from the host computer. Any following character sequence is then considered to be a terminal command and is automatically interpreted as such. Some of these commands control the position of the screen cursor. Special keyboard keys marked with arrows generate commands to move the cursor around the screen. Another command requests a report on the current line-column position of the cursor and the reported position can then be computer interpreted as needed for the simulated Anaplot interaction. A few subroutines make it possible to issue needed ESC commands from any Fortran program and to move manually the cursor on the screen, to report any chosen position to the program, to position the cursor from the program, to draw tables and display numbers or text in specific positions. A single ESC-code entered at any time while the analytical plotter program is active, is buffered and detected on the next entry to the PLOTTER routine so enabling an indefinite manual roaming with the cursor. Another ESC-key depression resumes subsequent PLOTTER functions, which then process information obtained from the operator's choice of the cursor position. The structure of the Anaplot version of the PLOTTER routine is as follows: control of simulation - console switches - X, Y, Z input - servo control - registers - optics - counters.

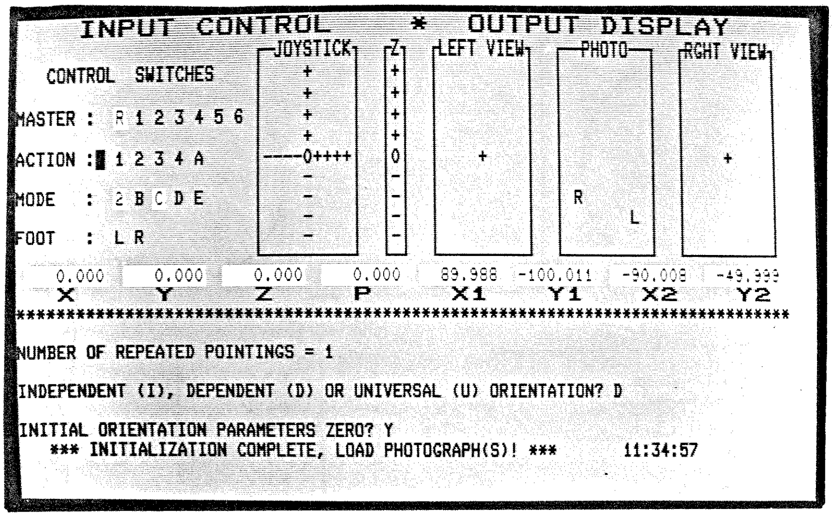

Fig. 7 Anaplot simulation by VT-100

As documented by the record in Figure 7, the initial phase of the Anaplot operation was just completed. Without going into specifics of the Anaplot control, some explanation of the displayed control is useful. One of the master switches, marked R (for RESET), is highlighted, which indicates that this pseudokey was cursor activated to initialize the procedure. Two other switches were also set on. In the row of mode switches key C called for a prestored list of conditions for the operation, while the activated '2' reflects the operator's choice of the right photo to be considered as a new, dependent picture in the process of bridging. Letters L and R in the 'PHOTO' area show the respective current positions of the measuring marks in the left and right photo stages needed for loading of photographs, as indicated in the record of the computer-operator dialog in the lower part of the screen. The optical fields do not show any displacement at this stage.

Another example in Figure 8 shows the situation when relative orientation is in progress during aerial triangulation. Activated master switch 2 characterizes the phase in progress. Mode switch 2 indicates that currently

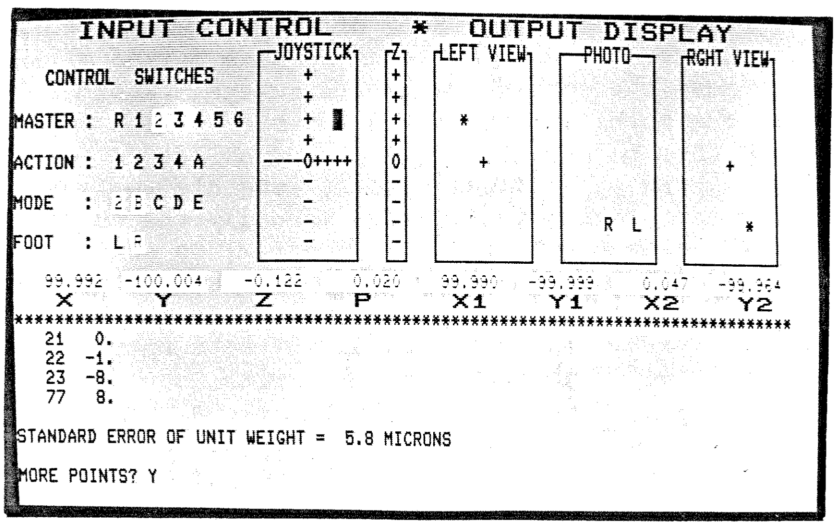

Fig. 8 Anaplot simulation by VT-100

it is the right photograph which is considered dependent, while mode B was chosen by the operator to display results of computations only in a condensed format. Currently, measurements are in progress by adding more points to a model already oriented. Right foot switch R is 'depressed' to perform measurements in a parallax mode, i.e. by moving only the dependent (in this case right) photo stage. The cursor in the joystick area indicates the direction in which the measuring mark is forced and the asterisks in both viewing fields show the current discrepancies of the fictitious optical details with respect to the measuring marks marked by '+' in the centre of the optical fields.

CONCLUSIONS

Successful feasibility test conducted at the National Research Council of Canada demonstrated how to simulate, in real time, the analytical plotter and its interface functions by software, using a suitable interactive video terminal. Analytical system simulations represent an important tool in developing software and in its testing without the presence of hardware components. This may be very useful in efforts pursuing either the standardization of on-line analytical systems or the development of a more universal system software.

REFERENCES

Jaksic, Z., 1983, Man-Machine Photogrammetric Systems and System Components, The Canadian Surveyor, 27(4): 308-319.

Jaksic, Z., 1980, Evaluation of Functional and Structural Characteristics of Analytical Plotters, International Archives of Photogrammetry, XXIII(B2): 106-116, ISP Hamburg.

PERS, 1982, Series of six papers on analytical plotters, Photogrammetric Engineering and Remote Sensing, 48(6): 891-940.