A time delay compensation technique for hybrid flight simulators

by V. W. INGALLS
The Boeing Company
Seattle, Washington

INTRODUCTION

A flight simulator or vehicle simulator of any type except those involving only a point mass, appears at first glance to be ideally suited for mechanization on a hybrid computer because of the difference in requirements of the rotational and translational equations of motion. Solutions to the rotational equations contain high frequencies, making the use of the analog computer best for solving these equations. The translational equations have mostly low frequency solutions but requires high accuracy so the digital computer is best. Also the digital computer is the best means of generating multi-variable functions such as aerodynamic coefficients. The equations to be solved and the conventional methods of mechanization of flight simulators are discussed by Fifer and Howe.

The division of the problem between the two halves of the hybrid computer is to solve the rotational equations mainly on the analog computer and the translational equations mainly on the digital computer. The rotational equations require multivariable functions so that part of the solution must be done digitally and because of the interaction of the translational equations with the rotational equations, part of the solution of the translational equations should be done by the analog computer. Because this division cannot completely isolate the rotational equations to the analog computer and the translational equations to the digital computer, several problems arise.

The problems

In the hybrid computer solution of differential equations several sources of error occur that are either not present or are easily compensated for in all digital or all analog computation. One of these sources of error, as reported by Miura and Iwata, is the delay introduced into a closed loop because of digital computation. The length of this delay is the time from an analog voltage being sampled and transferred to the digital computer until all computations using that value are completed and the results returned to the analog computer. This source of error has a counterpart in an all digital simulation but there are well known methods of reducing the error, in the all digital case. Some of these methods are directly applicable to hybrid computation, some can be used in modified form and some cannot be used at all. The most notable of the last group is the Runge-Kutta method. To use this method for hybrid computation would require the analog computer to make several trial solutions at the same instant of time which cannot be done. A number of papers have been written discussing various methods from the first two groups but most are only effective for linear problems.

Another source of error in a hybrid computer is data reconstruction following digital to analog conversion. This source of error, as well as some others, is discussed by Karplus. Theoretically, the output of a digital to analog converter is a string of impulses that are passed through some form of holding circuit. It is normally considered that reduction of the error by using any holding method more complicated than a simple zero order hold is not worth the problems created. This contention is not disputed but this paper will describe a method by which the effect of a higher order hold can be achieved without the actual hold circuit being present. One of the problems of using a zero order hold, which holds the value of the voltage constant between conversions, is that if the variable converted is a derivative to the integrated on the analog computer, which is often the case, especially in vehicle simulators, the integral can be determined algebraically at each sampling time. This means that although the integra-
tion is done by the analog computer, the result is the same as if it were done by the digital computer using the most elemental method of integration, i.e., rectangular or Euler. This being the case, there is some question as to the utility of using a hybrid computer for solving differential equations or at least for solving the equations of motion for a vehicle simulator.

A solution

The suggested solution, which is similar in some respects to that proposed by Gelman,5 to the problem of derivative functions being represented by a series of step functions is to rewrite the equations so that every loop can be closed without going through the digital computer. As an example, to check the method, the equation

\[ \frac{d^2y}{dt^2} = -ay | y | \]

was chosen because of several factors. These factors are, first, it is nonlinear; second, it has a solution that is periodic with constant amplitude and the period is functionally related to the amplitude; third, the period and amplitude can be determined analytically so errors can be checked; and last, it can be mechanized on an iterative differential analyzer simulating a hybrid computer. Actually this equation can be solved entirely by the analog computer, but the game was played assuming that the absolute value could not be found on the analog and thus the digital computer (or simulated digital computer) was needed.

The general form of this equation is

\[ \frac{d^2y}{dt^2} = f(y, \frac{dy}{dt}, t) \]

or in state equation form it is

\[ \frac{dx}{dt} = f(y, x, t) \]
\[ \frac{dy}{dt} = x \]

If \( x \) and \( t \) do not appear explicitly in the function \( f \), as in the example, the simplest way to rewrite the equation is to divide and multiply by \( y \) and mechanize it such that the function divided by \( y \) is generated by the digital computer and the multiplication by \( y \) is done by the analog. This mechanization might have some computation problems if \( y \) goes to zero but can be overcome if \( f \) also goes to zero for some \( y \). To do this, divide and multiply by \( y \) plus a constant, such that \( y \) plus the constant goes to zero when \( f \) goes to zero and define zero over zero as zero.

This mechanization will not completely eliminate the problems of the time delay or the use of the zero order hold. However the outputs of the analog integrators can no longer be computed algebraically at a sampling instant from the conditions at the previous sampling instant. Instead, a linear differential equation must be solved. The result of this manipulation, therefore, is to change a nonlinear differential equation to a linear equation with parameters that change at every sampling instant so as to approximate the original equation. This is also approximately the result of using a higher order hold in the data conversion equipment.

Nothing in the foregoing discussion does anything directly to overcome the time delay problem. The example will be used to describe the attack on this problem rather than a general case. Assume that the analog to digital converter, ADC, and digital to analog converter, DAC, operations occur simultaneously and repeat with a period \( \Delta t \), then the example equation as mechanized becomes

\[ \frac{d^2y(t)}{dt^2} = -a | y(t_1 - \Delta t) | y(t) \]

for \( t_1 \leq t < t_1 + \Delta t \). This means that the output of the DAC is delayed from the true function by one to two sampling periods, \( \Delta t \). To overcome this delay, either the function that is transferred by the DAC, \( -a | y(t_1 - \Delta t) | \), or the function transferred by the ADC, \( y \), can be predicted ahead one and one half periods using a truncated Taylor Series. The function \( y \) was chosen since the Taylor Series requires derivatives of the function and they are present in the analog computer. Therefore,

\[ y_p(t_1 + \frac{\Delta t}{2}) = y(t_1 - \Delta t) + \frac{dy}{dt}(t_1 - \Delta t) \]
If higher derivatives of the function to be predicted are available in the analog computer without differentiating they can be used to add more terms to the series.

The example equation was mechanized on an EASE 2100 Iterative Differential Analyzer as shown in the circuit diagram, Figure 1. The function switches allow for five different methods of solution. These are: all analog; hybrid with no compensation; hybrid with prediction only; hybrid with closed analog loop; and hybrid with both prediction and closed analog loop. Figure 3 shows the results of these five methods. The parameters and initial conditions were

\[ a = 16, \ y(0) = 0, \ \frac{dy}{dt}(0) = 2.8. \]

The \( \Delta t \) for hybrid with no compensation and hybrid with closed analog loop was .015 seconds and for the other two hybrid methods .15 seconds.

Although the solution using both compensations has a smaller change in amplitude with time than the solutions using the other methods, the error is noticeable and is dependent upon the sampling period. This last fact is particularly serious since the error tends to increase as the sampling period decreases.

To overcome this problem a further refinement was tried. Instead of dividing and multiplying \( f(y) \) by \( y \), a truncated Taylor Series expansion of \( f(y) \) about \( y_p \) was tried. The example equation, using only two terms of the series, becomes

\[ \frac{d^2 y}{dt^2} = -a \ | \ y_p | \ y_p - 2a \ | \ y_p | (y - y_p) \]

since the second term of the series is \( \frac{d^2 f(y_p)}{dy_p^2} (y - y_p) \) and the derivative of \( y \ | \ y \ | \) with respect to \( y \) is \( 2 \ | \ y \ | \).
Figure 2—Computer diagram for simulated hybrid solution of \( y = -a \frac{dy}{dt} \) by final method

Figure 3—Solutions to the equation \( y = -a \frac{dy}{dt} \) by all methods

Legend:
- --- NO COMPENSATION
- --- CLOSED LOOP ONLY
- --- PREDICTION ONLY
- --- PREDICTION & CLOSED LOOP
- --- ALL ANALOG AND PREDICTION & CLOSED LOOP BY SECOND METHOD
These variables are transferred to the digital computer and used to generate the functions \( f(X_1, X_2, \ldots, X_p, X_j, X_k, X_m, \ldots, X_n, t^p) \). The prediction of \( t \) is simply \( t^p = t + \frac{3}{2} \Delta t \). These variables are transferred to the digital computer and used to generate the functions \( f(X_1, X_2, \ldots, X_i, X_j, X_k, X_m, \ldots, X_n, t^p) \).

The computer diagram, again using the EASE to simulate a hybrid computer, is shown in Figure 2 and the results using the same parameters as before are shown in Figure 3 also. Other runs were taken using different sampling period, and for every run with a sample period shorter than the one shown, the solutions were identical as near as could be determined by eye. The runs made with larger sampling period showed a convergence so that in no case did the solution increase without bound.

Using the series expansion on \( f(y) \) not only reduces the error in solution to a negligible amount but also eliminates the possibility of division by zero and allows the method to be used for the general case where the derivative is a function of more than one variable. In this general case of a function of the form

\[
\frac{dx_i}{dt} = f(x_1, x_2, \ldots, x_n, t)
\]

assume \( x_i, x_j, x_k, x_m, \) and \( t \) are generated by the analog computer and the rest of the \( x \)'s are generated by the digital computer. Then \( x_{ip}, x_{jp}, x_{kp} \) and \( x_{mp} \) are generated using as many terms of a Taylor Series as there are derivatives available on the analog computer. The prediction of \( t \) is simply \( t^p = t_i + \frac{3}{2} \Delta t \). These variables are transferred to the digital computer and used to generate the functions \( f(x_1, x_2, \ldots, x_i, x_j, x_k, x_m, \ldots, x_n, t^p) \).

\[
\frac{\delta f}{\delta x_{ip}}, \quad \frac{\delta f}{\delta x_{jp}}, \quad \frac{\delta f}{\delta x_{kp}}, \quad \frac{\delta f}{\delta x_{mp}}, \quad \text{and} \quad \frac{\delta f}{\delta t_p}.
\]

These functions are then transferred back to the analog and combined to form the derivative

\[
\frac{dx_i}{dt} = f + \frac{\delta f}{\delta x_{ip}} (x_i - x_{ip}) + \frac{\delta f}{\delta x_{jp}} (x_j - x_{jp})
\]

\[+ \ldots + \frac{\delta f}{\delta t_p} (t - t_p)\]

where the predicted variables were either maintained on the analog computer by using memory integrators or were transferred back from the digital computer.

Another problem is the computation of initial conditions for the variables transferred from the digital computer to the analog computer. Since these variables are functions of the predicted variables produced by the analog which in turn are functions of the original variables, an iteration process must be performed. This iteration takes place while the analog computer is in Initial Condition mode so time is not changing, hence the prediction required is only a half step instead of a step and a half. This necessitates extra analog circuitry and some switching logic.

**Flight simulator mechanization**

The simulator described is being developed as a general purpose six degree of freedom simulation applicable to missiles, airplanes, or spacecraft; using a SDS 9300 digital computer and an EASE 2100 analog computer. The initial application is a short range guided missile and the description is based on this application.

The aerodynamic forces and moments for the missile and the partial derivatives of the forces and moments with respect to the aerodynamic attitude angles, \( \alpha \) and \( \beta \), fin deflection angles, \( \delta_1 \), \( \delta_2 \) and \( \delta_3 \) and body rotational rates, \( P, Q, \) and \( R \), are evaluated by the digital computer at the predicted sampling time. These values, including \( \alpha \), \( \beta \) and the \( \delta \)'s, are transferred through the interface to the analog computer where they are recomposed into the Taylor Series approximation of the continuous functions. The moments are combined with the moments of inertia and body angular rates to produce total body angular accelerations. These are integrated to give the body rates and also transformed by the Euler
angles to produce the angular accelerations of the Euler angles for prediction. The body rates are transformed by the Euler angles to produce the Euler angle rates, which are integrated to give the angles. The Euler angles, rates, and accelerations are combined, using Taylor Series, to produce predicted Euler angles. The sines and cosines of the predicted Euler angles are generated and transferred through the interface to the digital computer to be used to compute the predicted $\alpha$ and $\beta$.

The aerodynamic forces and the thrust are converted to linear accelerations and transformed from body axes components to earth axes components. The accelerations are integrated to produce the changes in velocity due to external forces. This integration uses an incremental or "hybrid integrator" circuit similar to that described by Wait, which operates until a fixed small increment of a velocity component has been accumulated and then generates an interrupt signal for the digital computer. The digital computer causes the integrator amplifier for that component to be reset to zero and adds the increment to the velocity component stored in digital memory and transfers the sum back to the analog computer. By using this circuit a more accurate solution can be achieved than by normal analog integration. The increments of velocity components and the linear accelerations are combined, using Taylor Series, to produce predicted velocity increments which are transferred through the interface to the digital computer.

The part of the velocity components returned from the digital computer are added to the increments to form total velocity components as continuous functions. These are transformed from earth axes back to body axes by the Euler angles and then transformed from rectangular to polar form yielding $\alpha$ and $\beta$ as continuous functions. These are needed for the reconstruction of the continuous forces and moments.

The flight control sub-system of the missile is simulated on the analog computer. It receives guidance commands in the form of commanded Euler angles from the digital computer and compares them with the actual Euler angles to produce an error. This error is used to determine the appropriate fin deflection angles commands. The simulated fin deflection actuators produce the actual deflection rates and angles. These are combined to give the predicted fin deflection angles that are transferred to the digital computer.

The digital computations are performed in the following order. At the beginning of each sampling period or frame, the ADC channels are sampled and their values converted and stored. All of these variables are the predicted versions of the corresponding analog variables so that there is a frame and one half lead at this instant. The rest of the variables stored in the digital computer at this time are one half frame ahead. The digital computer now integrates over one frame the equations defining the change in velocity components due to gravity and the rotating coordinate system, i.e. earth axes, and the equations defining the translational position of the vehicle. Since this integration requires a certain amount of computation time, all of the variables in the digital that are involved in the translation equations of motion now have somewhere between one and one and a half frame lead over the analog computer. The velocity components are now transformed by the predicted Euler angle to body axes and converted to polar form, resulting in predicted $\alpha$, $\beta$, and total relative vehicle velocity. The atmospheric properties are computed and the guidance equations are solved.

At this time, which is less than halfway through the frame all variables in the digital computer except the aerodynamic coefficients are updated to approximately one frame ahead of the analog computer. These variables include those for determining vehicle configuration changes such as thrust termination or staging or for determining the end of the run.

The tests are made and the time at which a change, if any, is to occur is determined by linear interpolation and the frame timer is set so as to stop the analog at the time of the change. The digital computer then goes back and recomputes the integrals up to the time of change. The reason for stopping the analog computer for just a configuration change is so that new data specifying that change can be read into the digital computer.

If no change conditions were met, the digital computer does the table look-ups of the aerodynamic coefficients as described previously and then waits until the fixed amount of time allowed for the frame has elapsed. The DAC values are then converted and after a very short delay to allow the transients on the analog computer due to the step changes in DAC outputs to settle, the ADC operation is done again and the next frame starts.

CONCLUSION
The six degree of freedom simulator discussed is not yet fully operational so a complete analysis of the accuracy and speed cannot be given. However, a three degree of freedom (one rotational and two translational) pitch plane only version of the simulator has been programmed using DES-1, the SDS 9300 digital analog simulation language, to represent a hybrid computer. The results from this simulator
indicate that the frequency and damping of the rotational motion agrees to within an acceptable limit with that calculated from a linearized model and that the accuracy of trajectory computations is as good as that from an all digital simulation. The frame time necessary to achieve this was 1/10 to 1/12 of the period of the highest frequency appearing at the interface.

REFERENCES

1 S FIFER
   *Analog computation*

2 R M HOWE
   *Coordinate systems for solving three dimensional flight equations*

3 T MIURA J IWATA
   *Effects of digital execution time in a hybrid computer*
   Spartan Washington DC 1963

4 W J KARPLUS
   *Error analysis of hybrid computer systems simulation Vol 6 pp 120-136 February 1966*

5 R GELMAN
   *Corrected inputs: a method for improved hybrid simulation*
   Spartan Washington DC 1963

6 J V WAIT
   *A hybrid analog-digital differential analyzer system*
   Spartan Washington DC 1963