Optimal Digital Simulation of Aircraft via Random Search Techniques

Guy O. Beale*
Babcock and Wilcox, Lynchburg, Va.
and
Gerald Cook†
University of Virginia, Charlottesville, Va.

This paper discusses a technique for the development of a discrete time integration operator to be used in the simulation process. The integration operator can be optimized for a particular system subjected to a set of specified inputs. The class of systems being investigated are those which can be represented by a set of state equations. A discrete time integration operator with certain free parameters is hypothesized. An adaptive random search optimization (ARSO) technique is used to find the optimum values for these parameters. Examples are presented to show the effectiveness of this technique.

I. Introduction

REAL-TIME digital simulation of physical systems has received attention for a number of years.[1,2] This paper discusses a technique for the development of a discrete time integration operator to be used in the simulation process. The integration operator will be optimized for the particular system being simulated, subjected to a set of specified inputs. The types of systems being investigated are those which can be represented by a set of state equations

$$\dot{x}=f(x,\ u) \tag{1}$$

where x is the $n\times 1$ state vector, u is the $r\times 1$ control vector, and f is the set of typically nonlinear n functions.

II. Integration Operator

Figure 1a is a block diagram of the mathematical relations in Eq. (1). The vectors x and u are acted upon by the functional relations $f(x,\ u)$, producing the vector $\dot{x}$, which is then integrated to produce the state vector x. Figure 1b is a block diagram of a discrete approximation to the continuous time system. The control vector u is assumed to be sampled at a uniform rate, producing the input samples $u(k)$. The equations

$$\dot{x}(k)=f[x(k),\ u(k)] \tag{2}$$

are in the same form as those representing the continuous time system. For example, if

$$\dot{x}_1=\cos x_2+\sin x_3+u$$

then

$$\dot{x}_1(k)=\cos x_1(k)+\sin x_3(k)+u(k)$$

The function $F(z)$ in Fig. 1b represents the discrete integration operator. For the simulation to be realizable, the denominator of $F(z)$ must be of a higher power in z than the numerator.[1] This can be shown to be correct, as follows. Let $F(z)$ be

$$F(z)=\frac{T}{2}\frac{z+1}{z-1}=\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}$$

which is the familar Tustin substitution operator for continuous integrator $1/s$. Expressing $F(z)$ as the ratio of the integration operator's output to its input yields

$$x(z)/\dot{x}(z)=(T/2)(1+z^{-1})/(1-z^{-1})$$

$$(1-z^{-1})x(z)=T/2(1+z^{-1})\dot{x}(z)$$

$$x(z)=z^{-1}x(z)+T/2(1+z^{-1})\dot{x}(z)$$

$$x(k)=x(k-1)+T/2[\dot{x}(k)+\dot{x}(k-1)] \tag{3}$$

The calculation of the state at time k is seen to depend upon its derivative at time k. However, from the state equations, the derivative at time k is a function of the state at time k, Eq. (2). This leads to an equation of the form

$$x(k)=x(k-1)+T/2\{f[x(k),\ u(k)]+f[x(k-1),u(k-1)]\}$$

Since the functions $f(x,\ u)$ are typically nonlinear, the equation cannot be solved by factoring $x(k)$ out of the right-hand side of the equation. Therefore, Eqs. (3) represent an unrealizable simulation. If the integration operator had been chosen as

$$F(z)=\frac{T}{2}\left(\frac{z+1}{z-1}\right)z^{-1}=\frac{T}{2}\frac{z^{-1}+z^{-2}}{1-z^{-1}}$$

then

$$x(k)=x(k-1)+T/2[\dot{x}(k-1)+\dot{x}(k-2)]$$

which is a realizable simulation.

Adding the z^{-1} term to the Tustin operator, however, degrades its performance. It has been shown, though, that to be closed-loop realizable, the power of the denominator must exceed that of the numerator. What is desired is a closed-loop realizable operator which can be optimized for a particular system being driven by a set of known inputs. To this end, a

Received Aug. 30, 1977; revision received Dec. 15, 1977.
Index category: Analytical and Numerical Methods.
*Senior Engineer.
†Professor of Electrical Engineering.

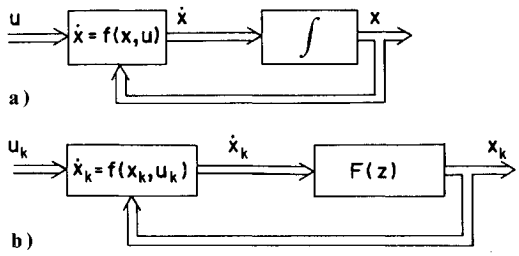

Fig. 1 Schematic representation of time state equation. a) Continuous, b) Discrete.

discrete-time integration operator of the following form is chosen.

$$F(z) = \frac{T\sum_{j=0}^{N} \lambda_j z^j}{z^N(z-1)} = \frac{T(\lambda_0 + \lambda_1 z + \ldots + \lambda_N z^N)}{z^{n+1} - z^N}$$

$$F(z^{-1}) = \frac{T[\lambda_N z^{-1} + \lambda_{N-1} z^{-2} + \ldots + \lambda_2 z^{-N} + \lambda_0 z^{-(N+1)}]}{1 - z^{-1}}$$

where T is the sampling period and the λ's are a set of free parameters, the values of which are to be optimized. This operator yields a realizable simulation, since the power of the denominator is always one greater than that of the numerator. The pole at $z=1$ corresponds to a pole at the origin in the complex s plane, and the Nth order pole at the origin in the z plane corresponds to an Nth order pole at negative infinity in the s plane.[3] Therefore, the transient response of the poles added at $z=0$ to make the operator closed-loop realizable will decay quickly. The state equations are now of the form

$$\dot{x}(k) = f[x(k),\ u(k)]$$

$$x(k+1) = x(k) + T[\lambda_N \dot{x}(k) + \lambda_{N-1}\dot{x}(k-1) + \ldots + \lambda_0 \dot{x}(k-N)] \quad (4)$$

Equation (4) can be thought of as a polynomial approximation to the value of a function at point $(k+1)$ based on its value at point (k) and the value of its derivative at point (k) and preceding.

The free parameters in $F(z)$ are optimized using an idealized model form of a model reference adaptive control system.[4] Figure 2 is a block diagram of this configuration. An exact, or nearly exact, solution to the system's differential equations represents the ideal model of the system. In this case, a fourth-order Runge-Kutta numerical integration of the state equations is used with a sufficiently small step size to guarantee an accurate solution. In some cases, this step size is one-tenth the sampling period of the discrete simulation. The discrete simulation, including the parameters to be optimized, represents the controlled process. A set of inputs is applied to the model and the process, and the error at each sample time is squared and summed. At the end of one run, the free parameters are perturbed under the control of an optimization technique, and the sequence is repeated. This continues until the error of the digital simulation has reached its minimum.

III. Optimization Technique

The perturbation of parameters in $F(z)$ is controlled by an adaptive random search optimization (ARSO) technique. Random perturbation methods have been shown to solve a large class of optimization problems faster than gradient techniques when the number of unknown parameters exceeds four.[5-7] In addition, the convergence time has empirically

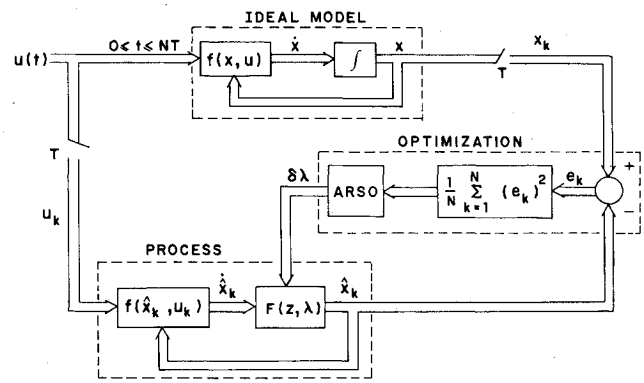

Fig. 2 Schematic representation of optimization configuration.

been shown to increase linearly with the number of unknown parameters, rather than exponentially or quadratically.[7]

In the ARSO technique, the mean and variance of a uniformly distributed random variable are adaptively selected for each unknown parameter based on recent successful experiments. In this way, the step size and direction are controlled by previous successful perturbations. Adaptive step size has been shown by Schumer and Steiglitz[7] to be a powerful technique in multidimensional problems without ridges or valleys. Adding adaptive step direction should tend to broaden the range of applicability.

The performance index used with ARSO is a vector-valued one; that is, one which requires simultaneous minimization of all components.[5,8-10] This allows consideration of several cost functions, such as integral square error, minimum energy, etc., at the same time. In the nonlinear problem to be studied, aircraft dynamics, the mean square error for each of the state variables, is used as a component in the performance index. That is,

$$J_i = \left[1/M \sum_{j=1}^{M} (x_{ij} - x_{ij}^*)^2\right]^{1/2}$$

and

$$J = [J_1, J_2, \ldots, J_n]T$$

where M is the number of sample periods per trial, x_{ij} is the approximate value of the ith state variable at the jth sample time, and x_{ij}^* is the exact value. For a trial to be considered a success, at least one component of J must be reduced, and no component may increase in value.

The unknown parameters are perturbed in the following manner:

$$\lambda_i(j+1) = \lambda_i^* + \delta\lambda_i(j+1) \quad (5)$$

where $\lambda(j+1)$ is the new value of the ith parameter; λ_i^* is the "best-to-date" value of the ith parameter, that is, the value of λ_i when the minimum-to-date value of the J vector was calculated; and $\delta\lambda_i(j+1)$ is the random perturbation for the ith parameter. This is equivalent to the perturbation scheme shown in the following.[7]

$$\lambda_i(j+1) = \lambda_i(j) - a(j)\,\delta\lambda_i(j) + \delta\lambda_i(j+1)$$

where

$$a(j) = 0 \quad \text{if } J_{(j)} < J^*_{(j-1)}$$

$$1 \quad \text{if } J_{(j)} > J^*_{(j-1)}$$

$J_{(j)}$ is the performance vector of the jth trial, and $J^*_{(j-1)}$ is the smallest performance vector obtained through $(j-1)$ trials. The coefficient $a(j)$ is used to negate the effect of an un-

successful trial. The perturbation is calculated as

$$\delta\lambda_i(j) = \mu_i(j) + \sqrt{3\sigma_i^2(j)}\,[2\text{RND}(0) - 1] \tag{6}$$

where $\mu_i(j)$ is the current value for the mean of the ith random variable, $\sigma^2(j)$ is the current value for the variance, and RND(0) is a uniformly distributed random variable on the interval [0, 1]. Equation (6) produces a random number from a uniformly distributed random variable with mean μ and variance σ_2. Stability considerations may place constraints on the parameter calculated in Eq. (5). If a particular perturbation places a value outside its limit for stability, the value may be moved deterministically inside the limit or another random perturbation may be tried. In simulating complex systems, however, it may be difficult to determine the stability limits for the coefficients a priori. In this case, one or more of the state variables may be monitored during the simulation, and if they exceed reasonable values, the trial may be aborted. This saves computation time and may prevent the entire program from being terminated due to overflow.

When a particular trial is successful, that is,

$$J_i \le J_i^* \qquad 1 \le i \le n$$

where J_i^* is the minimum value of the ith element of J, the means and variances of the random variables are updated. The mean is calculated as

$$\mu_i(j) = \lambda_i^* - m_i \tag{7}$$

where

$$m_i = \frac{\lambda_{i(j)}}{2} + \frac{\lambda_{i(j-1)}}{4} + \frac{\lambda_{i(j-2)}}{8} + \frac{\lambda_{i(j-3)}}{16} + \frac{\lambda_{i(j-4)}}{16} \tag{8}$$

The λ_i's are past values of λ_i^* that is, previous values of the best-to-date parameters with $\lambda_i(j)$ being the current value of λ_i^*. The mean is thus seen to be the difference between the current best value and a weighted sum of previous best values. If all of the λ_i's lie on the same side of the locally optimum parameter value, Eqs. (7) and (8) will always produce a mean which tends to move the parameter value toward that optimum. If the λ_i's are on both sides of the optimum value, the mean at times may have the wrong algebraic sign. The magnitude of the mean does adjust itself to the differences between successfive values of λ_i^*, with the most recent differences being weighted most heavily.

The variance for the distribution is determined using the following argument. The perturbation for each parameter is a random variable with a uniform probability density function with mean μ_i and variance σ_i^2. Relating these moments to the end points of the function (a, b) yields

$$\mu_i = (b_i + a_i)/2, \qquad \sigma_i^2 = (b_i - a_i)^2/12$$

If the mean and one end point are known, the other end point and variance can be calculated. That is, if μ and a are known, then

$$b_i = 2\mu_i - a_i, \qquad \sigma_i^2 = (\mu_i - a_i)^2/3$$

and if μ and b are known, then

$$a_i = 2\mu - b_i, \qquad \sigma_i^2 = (b_i - \mu_i)^2/3$$

Relating the end points of the density function to the parameter values yields the expression for the variance

$$\sigma_i^2(j) = [\mu_i(j) - {}^{\lambda}i(j-1) + {}^{\lambda}i(j)]^2/3 \tag{9}$$

Therefore, Eq. (9), combined with the mean from Eqs. (7) and (8), provides the necessary data for computing the perturbations. The variance in Eq. (9) will provide a wide enough range in perturbations to allow a reduction in the performance index even when the mean has the incorrect sign. When a large number of consecutive failures are generated, the means and variances are set to deterministic values. This allows the search technique to more fully explore the parameter space about which little is known beforehand. If no improvements are obtained after a set number of failures, the search is terminated or restarted with a different set of initial conditions.

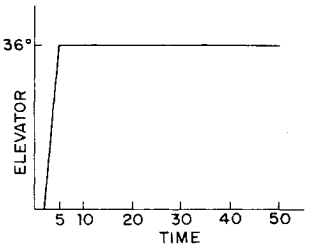

Fig. 3 Control input for sixth-order model.

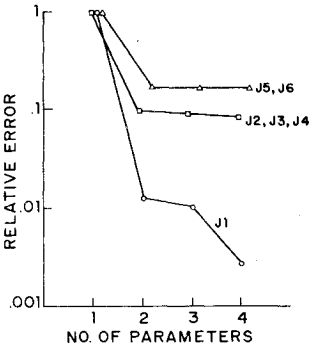

Fig. 4 Relative change in errors.

IV. Results

The ARSO technique has been applied to the problems of simulating an aircraft. Since, at the present time, the primary purpose is to develop the optimization technique, several assumptions have been made. The aircraft control is operating open loop, that is, the pilot and autopilot are not being considered. The control surfaces are given a particular setting, and the aircraft dynamics alone determine the resultant trajectory. Also, no random disturbances are considered. The state equations contain no stochastic forcing functions. Motion in the vertical plane only is initially considered. The equations of motion are:

$$\begin{aligned}
\dot{x} &= V\cos(\gamma)\\
h &= V\sin(\gamma)\\
\dot{v} &= -g\sin(\gamma) - (d/m)v^2 + (T/m)\cos(\alpha)\\
\dot{\gamma} &= (1/v)\,[-g\cos(\gamma) + (\ell/M)v^2 + (T/m)\sin(\alpha)]\\
\dot{a} &= \omega\\
\dot{\omega} &= 1.311\,u - 0.806\,\omega - 1.311\,\alpha
\end{aligned} \tag{10}$$

where x is the horizontal displacement, h the vertical displacement, v the total velocity, γ the flight path angle, α the angle of attack, u the control elevator deflection, and ω the time rate change of angle of attack.

Each of the six states generates an element of the performance index. The mean-squared error between the approximate value and a value obtained by Runge-Kutta integration is used. Figure 3 shows the control history used for the simulation. Although this may not be a realistic control input for an aircraft, it does exercise the model enough for the nonlinearities in Eqs. (10) to be felt in a short amount of time. Therefore, for the purpose of this research, this input seems to be justified.

The results of ARSO optimization using one, two, three, and four parameters are shown in Table 1a. The sample

Table 1a Root mean square errors in integrating sixth-order model Using ARSO

	ARSO-1	ARSO − 2	ARSO − 3	ARSO − 4
J1	126	1.68	1.36	0.303
J2	3.4	0.282	0.268	0.233
J3	0.15	1.31×10^{-2}	1.25×10^{-2}	1.16×10^{-2}
J4	4.1×10^{-4}	4.39×10^{-5}	4.34×10^{-5}	4.26×10^{-5}
J5	7.0×10^{-2}	1.18×10^{-2}	1.18×10^{-2}	1.17×10^{-2}
J6	8.9×10^{-2}	1.32×10^{-2}	1.31×10^{-2}	1.30×10^{-2}

ARSO − 1: $\lambda_0 = 1.00722$
ARSO − 2: $\lambda_0 = 1.50358$, $\lambda_1 = -0.503668$
ARSO − 3: $\lambda_0 = 1.50247$, $\lambda_1 = -0.504014$, $\lambda_2 = 1.47173\times10^{-3}$
ARSO − 4: $\lambda_0 = 1.49938$, $\lambda_1 = -0.502489$, $\lambda_2 = 1.46801\times10^{-3}$, $\lambda_3 = 1.98933\times10^{-3}$

Table 1b Root mean square errors in integrating sixth-order model using classical techniques

	Forward difference, $T=0.5$ s	Delayed Tustin, $T=0.25$ s	Milne-Reynolds, $T=0.25$ s
J1	3.89	3.99	0.011
J2	3.7	3.4	0.014
J3	0.174	0.178	4.12×10^{-11}
J4	3.32×10^{-4}	3.32×10^{-4}	4.12×10^{-11}
J5	6.9×10^{-2}	6.4×10^{-2}	2.6×10^{-8}
J6	7.8×10^{-2}	7.3×10^{-2}	1.1×10^{-7}

Forward difference: $\lambda_0 = 1$
Delayed Tustin: $\lambda_0 = 0.5$, $\lambda_1 = 0.5$

period for each of these is 0.5 s. This large value for the sample period is used to test the effectiveness of the AROS technique in producing stable simulations with larger than normal sample periods. The Runge-Kutta integration, which is used to determine the accuracy of the simulation, uses a sample period of 0.05 s. Shown in Table 1b are the results using the forward difference operator, the delayed Tustin operator, and the Milne-Reynolds predictor-corrector method. The Milne-Reynolds and delayed Tustin results are for a sampling period of 0.25 s, as both of them are unstable for $T=0.5$ s. The total amount of computation time per second of simulation time was four times greater for Milne-Reynolds than for the discrete operators obtained via AROS.

As can be seen from Table 1b, the values for each of the performance vector elements decrease as additional parameters are introduced into the integration operator. The most significant decrease occurs for the aircraft range J1 in going from one to two parameters. The aircraft range is a very sensitive function of parameter value when a single parameter is used. The error reaches its minimum value, 3.9, for a parameter value of unity; i.e. the forward difference operator. The ARSO-1 solution is worse on this component of the performance vector but better on some others. The two solutions are incomparable and point out the dependence of the final solution on the initial guess for the parameter values. In going from ARSO-1 to ARSO-2, J1 is reduced by two orders of magnitude, J2, J3, and J4 are reduced by one order of magnitude, and J4 and J5 are reduced by a factor of five. Very little further improvement is obtained with the addition of a third parameter. A large decrease in range error does result when four parameters are used.

Figure 4 illustrates in a qualitative sense the effect of adding parameters to the integration operator. One curve represents J1, one represents J2, J3, and J4, and one curve represents J5 and J6. Normalizing the cost values to unity for one parameter, the curves show the large decrease in the cost function when a second parameter is added and the much smaller improvements realized with additional parameters. Since very little improvement is obtained with three rather than two parameters, it seems reasonable to use ARSO-2. If additional accuracy is required, ARSO-4 reduces the range

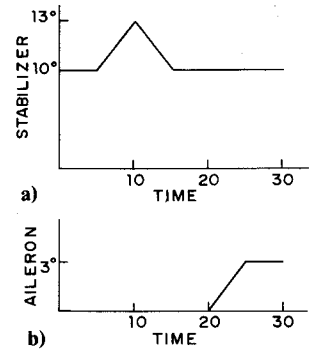

Fig. 5 Control inputs for twelfth-order model.

error by 82% relative to ARSO-2 and by 78% relative to ARSO-3. The error in aircraft altitude is reduced by 17% relative to ARSO-2 and by 13% relative to ARSO-3. Although the trend is not confirmed by this small sample, the data indicate that a greater improvement is obtained in going from an odd to an even number of parameters than vice versa. This implies that using a particular operator with an even number of parameters would yield approximately the same results and require less computation time than an operator with one additional parameter.

The simulations were tested with two other control functions of the same general type as the first. All errors remained of the same order of magnitude, indicating that performance of the simulation would not be significantly degraded when the input changed.

To further test the effectiveness of the ARSO technique, the optimum set of parameters for the discrete time simulation of a twelfth-order nonlinear model of an aircraft were determined. The aerodynamic coefficients for lift, drag, side force, pitch, yaw, and roll moment were all approximated by linear functions of the controls and states for convenience. However, all the other nonlinearities, such as the cross-coupling terms, the dependence of the forces and moments on the square of velocity, and also the trigonometric functions, such as those in Eqs. (10), were retained. This model is presented in detail in Ref. 11.

The control used for this simulation involved stabilizer and aileron deflections, as shown in Fig. 5. This provides motion in all three translational and rotational axes. The simulation time interval is taken to be 30 s, and the sample period is 0.5 s. A longer simulation interval was not used due to the computation time and cost involved. The authors feel that the 30-s simulation provides an adequate exercise for the optimization procedure.

Optimizations with one and two free parameters were carried out. A third optimization, using the results of ARSO-2 as initial conditions, was attempted, but no improvement in the value of the cost function was obtained. A 12-element vector performance index was used during the optimization.

The ARSO results are compared with the forward difference operator, the delayed Tustin operator, and a second-order operator with values near the optimum values of ARSO-2. The equations for these three operators are

$$x(k+1) = x(k) + T\dot{x}(k)$$

$$x(k+1) = x(k) + T[(1/2)\dot{x}(k) + (1/2)\dot{x}(k-1)$$

$$x(k+1) = x(k) + T[1.5\dot{x}(k) - 0.5\dot{x}(k-1)$$

respectively. Comparison of the results with the Milne-Reynolds predictor-corrector method was not done in this example. It is assumed that this method will produce an extremely accurate simulation similar to that obtained in the previous example. The penalty for this accuracy is the increased computation time. Recall that, for the previous example, the Milne-Reynolds method required more than four times as much computation time as ARSO results. In

Table 2a Root mean square errors in integrating twelfth-order model using ARSO

	ARSO − 1	ARSO − 2
J1	9.12	0.1787
J2	4.06×10^{-3}	2.07×10^{-3}
J3	2.38×10^{-4}	1.29×10^{-5}
J4	1.71×10^{-4}	4.74×10^{-6}
J5	9.30×10^{-3}	2.14×10^{-4}
J6	7.97×10^{-5}	1.17×10^{-6}
J7	0.3880	0.3732
J8	5.25×10^{-2}	1.21×10^{-2}
J9	0.3883	0.3732
J10	147.6	32.79
J11	149.3	2.42

ARSO − 1: $\lambda_0 = 1.02052$
ARSO − 2: $\lambda_0 = 1.59111$, $\lambda_1 = -0.596767$

Table 2b Root mean square errors in integrating twelfth-order model using classical techniques

	Forward difference	Delayed tustin ($T=0.25$s)	Modified ARSO
J1	15.2	15.0	1.22
J2	6.5×10^{-3}	6.43×10^{-3}	5.86×10^{-4}
J3	3.24×10^{-4}	3.20×10^{-4}	3.4×10^{-5}
J4	2.57×10^{-4}	2.54×10^{-4}	2.26×10^{-5}
J5	1.48×10^{-2}	1.46×10^{-2}	1.27×10^{-3}
J6	1.08×10^{-4}	1.07×10^{-4}	1.25×10^{-5}
J7	0.3889	0.3888	0.3788
J8	8.64×10^{-2}	8.53×10^{-2}	1.19×10^{-2}
J9	0.3894	0.3893	0.3788
J10	68.5	67.7	9.92
J11	0.7795	0.7668	0.3086
J12	241.4	238.6	20.3

Forward difference: $\lambda_0 = 1$
Delayed Tustin: $\lambda_0 = 0.5$, $\lambda_1 = 0.5$
Modified ARSO: $\lambda_0 = 1.5$, $\lambda_1 = -0.5$

real-time simulation, this additional time may not be available.

Table 2a lists the results for the ARSO-1 and ARSO-2 optimizations, where the J_i represent the rms errors in the respective state variables relative to the Runge-Kutta solution.

As with the results of the previous example, all elements of the ARSO-2 performance vector are smaller than the corresponding elements for the ARSO-1 vector. Elements J1, J5, J6, and J12 are reduced by factors ranging from 43 to 61; elements J3 and J4 are reduced by factors of 18 and 36, respectively.

Table 2b lists the rms errors for the forward difference, delayed Tustin, and modified ARSO operators. This last operator is chosen with values near the optimum parameter values for ARSO-2. It is interesting to note that this set of parameter values corresponds to the Adams-Bashforth second-order integration algorithm. Since the ARSO-2 results for both the vertical plane simulation and this simulation converge toward these values, the modified ARSO provides a good comparison for the optimization results. As seen in Table 2b, the ARSO-2 results are better than the forward difference and delayed Tustin results in each entry; in some cases, by as much as two orders of magnitude. It also should be noted, that the sample period for the delayed Tustin operator is 0.25 s for stability reasons. Thus, twice as many points are required to cover the same simulation interval. The ARSO-1 optimization produces smaller errors in each case, except for the longitudinal range error. As mentioned earlier, this error is a sensitive function of parameter value in the single parameter case; however, it is one of the least significant variables in a simulation. The roll and yaw angle errors for all five of the integration operators considered are approximately the same value.

The modified ARSO operator has smaller errors than ARSO-2 for angle of attack, pitch angle, and range. The other elements have larger values, with velocity and altitude errors being nearly an order-of-magnitude larger than ARSO-2.

The authors feel that the reason for not being able to obtain improvement using an ARSO-3 optimization is due to the small errors produced by ARSO-2. Since the pitch and yaw rates and the side slip angle already have such small errors, it would of course, be difficult to reduce the other errors without allowing these two errors to increase somewhat. The vector performance index does not allow this.

Comments on Computation

Various computers were used during different stages of ARSO development. Initial development was done on a 16-bit time-shared minicomputer. While this provided convenience during early strategy development, it was much too slow for actual simulation. A dedicated 16-bit minicomputer was then tried. For the sixth order, vertical plane aircraft model, approximately 8s were required to solve the dynamic equations, compute the performance index, compare this index with the best-to-date value, and perturb the coefficients of the integration operator. Although this was a large improvement over the time-shared system, it was still too slow to allow a large number of perturbations.

For the results presented in Ref. 11, a CDC Cyber 76 mainframe computer was used. For the twelfth-order model, approximately 30,000 perturbations could be done in 15 min of CPU time. This type of computation power is necessary for optimizing the simulations of high-order, nonlinear systems.

V. Conclusions

The use of an ARSO technique for determining the optimum parameters for a discrete integration operator has been presented, along with the notion of a vector performance index. This procedure has been applied to an aircraft modeled by high-order, complex, nonlinear differential equations. The control sequence used exercised all modes of behavior during the simulation. The ARSO technique has produced a discrete time simulation with a large sample period, good accuracy, and short computation time, demonstrating the effectiveness of the technique.

References

[1] Sage, A.P., "A Technique for the Real Time Digital Simulation of Non-Linear Control Processes," *Proceedings of Region III IEEE Conference,* April 1966.

[2] Sage, A.P. and Smith, S.L., "Real Time Digital Simulation for Systems Control" *Proceedings of the IEEE,* Vol. 54, Dec. 1966, pp. 1802-1811.

[3] Kuo, B.C., "Analysis and Synthesis of Sampled Data Control Systems," Prentice-Hall, New York, 1963.

[4] Beck, M.S., "Adaptive Control – Fundamental Aspects and Their Application," *Proceedings of 1st Annual Advanced Control Conference,* Dun-Donnelly Publishing, Purdue University, April. 29-May 1, 1974.

[5] Andrews, M. and Korn, G.A., "Hybrid Computer Method for Functional Optimization," *IEEE Transactions on Computers,* Vol. C-24, No. 10, Oct. 1975, pp. 958-965.

[6] Korn, G.A. and Kosako, H., "A Proposed Hybrid Computer Method for Functional Optimization," *IEEE Transactions on Computers,* Feb. 1970, pp. 149-152.

[7] Schumer, M.A. and Steiglitz, K., "Adaptive Step Size Random Search," *IEEE Transactions on Automatic Control,* Vol. AC-13, No. 3, June 1968, pp. 270-276.

[8] Zadeh, L.A., "Optimality and Non-Scalar Valued Performance Criteria," *IEEE Transactions on Automatic Control,* Vol. AC-8, No. 1, Jan. 1963, p. 59.

[9] Geering, H.P. and Athans, M., "The Infimum Principle," *IEEE Transactions on Automatic Control,* Vol. AC-19, No. 5, Oct. 1974, pp. 485-493.

[10] Athans, M. and Geering, H.P., "Necessary and Sufficient Conditions for Differential Nonscalar-Valued Functions to Attain Extrema," *IEEE Transactions on Automatic Control,* Vol. AC-18, No. 2, April 1973, pp. 132-138.

[11] Beale, G.O., "Optimal Aircraft Simulator Development by Adaptive Random Search Optimization," Ph.D. Dissertation, University of Virginia, May 1977.