

United S
Alexander

165123

X65159 ✓

X65007 ✓

[11] 3,748,440

[45] July 24, 1973

- [54] BALLISTIC TRAJECTORY COMPUTER
[75] Inventor: Thomas F. Alexander, Rolling Hills Estates, Calif.
[73] Assignee: Hughes Aircraft Company, Culver City, Calif.
[22] Filed: Nov. 20, 1969
[21] Appl. No.: 886,600

Analyzers. Part IV of IRE Convention Record July 1954 p. 87-97.

Primary Examiner—Felix D. Gruber
Attorney—James K. Haskell and Walter J. Adam

- [52] U.S. Cl. 235/61.5 R, 235/61.5 E, 235/150.31, 235/152
[51] Int. Cl. G06f 15/58
[58] Field of Search 235/61.5 R, 61.5 E, 235/152, 183, 184, 150.31; 89/1.5 E, 41 E, 41 EA

[56] **References Cited**
UNITED STATES PATENTS

- | | | | |
|-----------|---------|--------|------------|
| 2,833,470 | 5/1958 | Welty | 235/61.5 E |
| 2,947,474 | 8/1960 | Vance | 235/61.5 E |
| 3,063,047 | 11/1962 | Steele | 235/152 |
| 3,246,130 | 4/1966 | Rubin | 235/61.5 R |

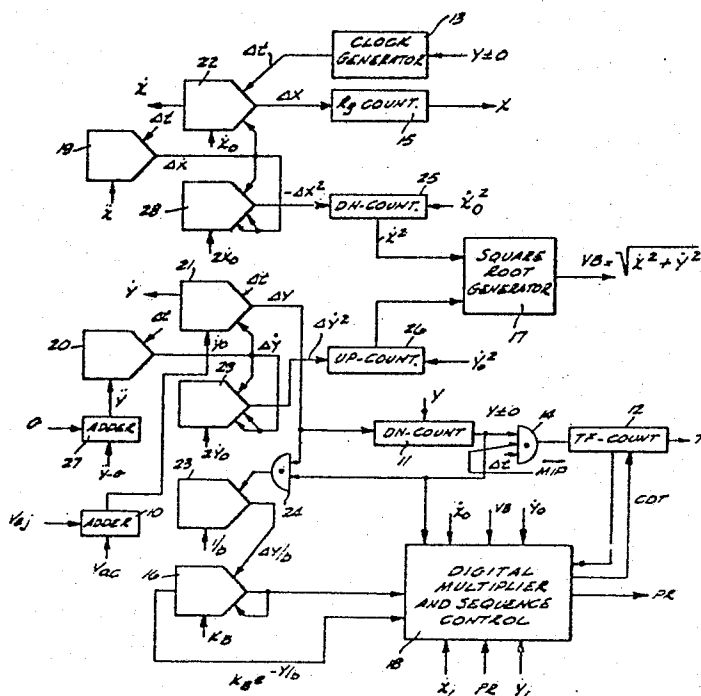
OTHER PUBLICATIONS

Braun: Design Features of Current Digital Different

[57] **ABSTRACT**

A real-time system is provided for calculating in less than about 40 msec. of actual time the ballistic time of fall and range of a weapon from a known altitude above a target using weighted clock pulses for digital integrators in multi-loop function generators. Digital integrators of the system are first used to generate initial conditions for the solution of simultaneous second order non-linear differential equations of motion developed from the geometry and physics of the ballistic trajectory problem assuming a cool exponential atmosphere. The multi-loop function generators then solve the simultaneous differential equations with interruptions to recalculate initial conditions as necessary.

13 Claims, 8 Drawing Figures



1 capsule ballistic
2) bomb

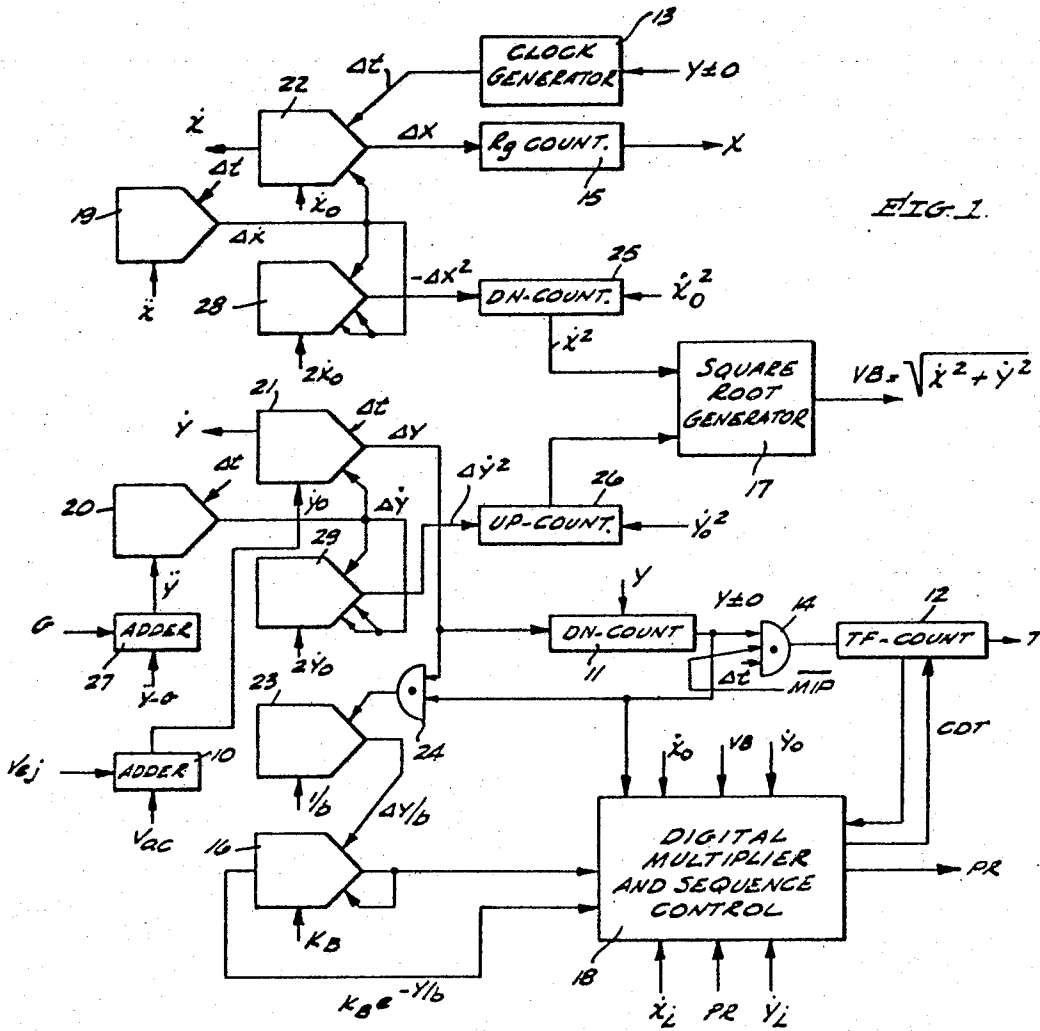
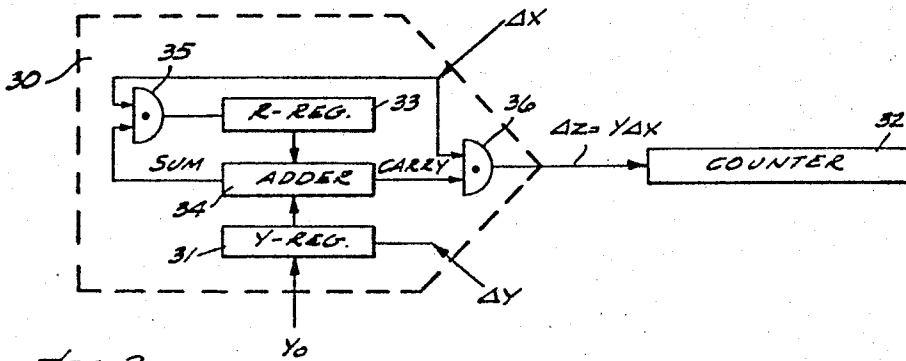


FIG. 2.

INVENTOR.
 THOMAS F. ALEXANDER.
 BY *J. K. Haskell*
 ATTORNEY.

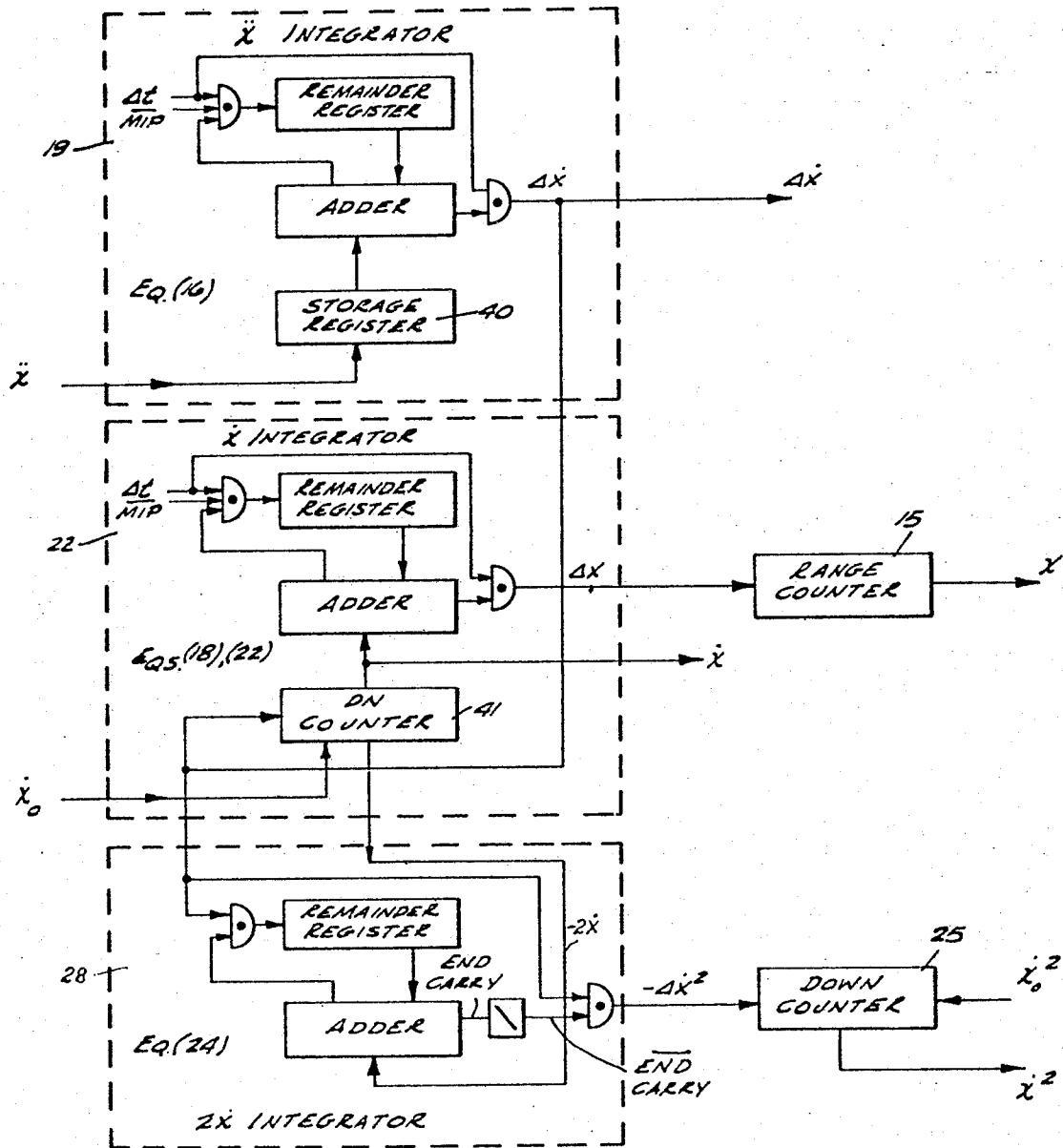


FIG. 3.

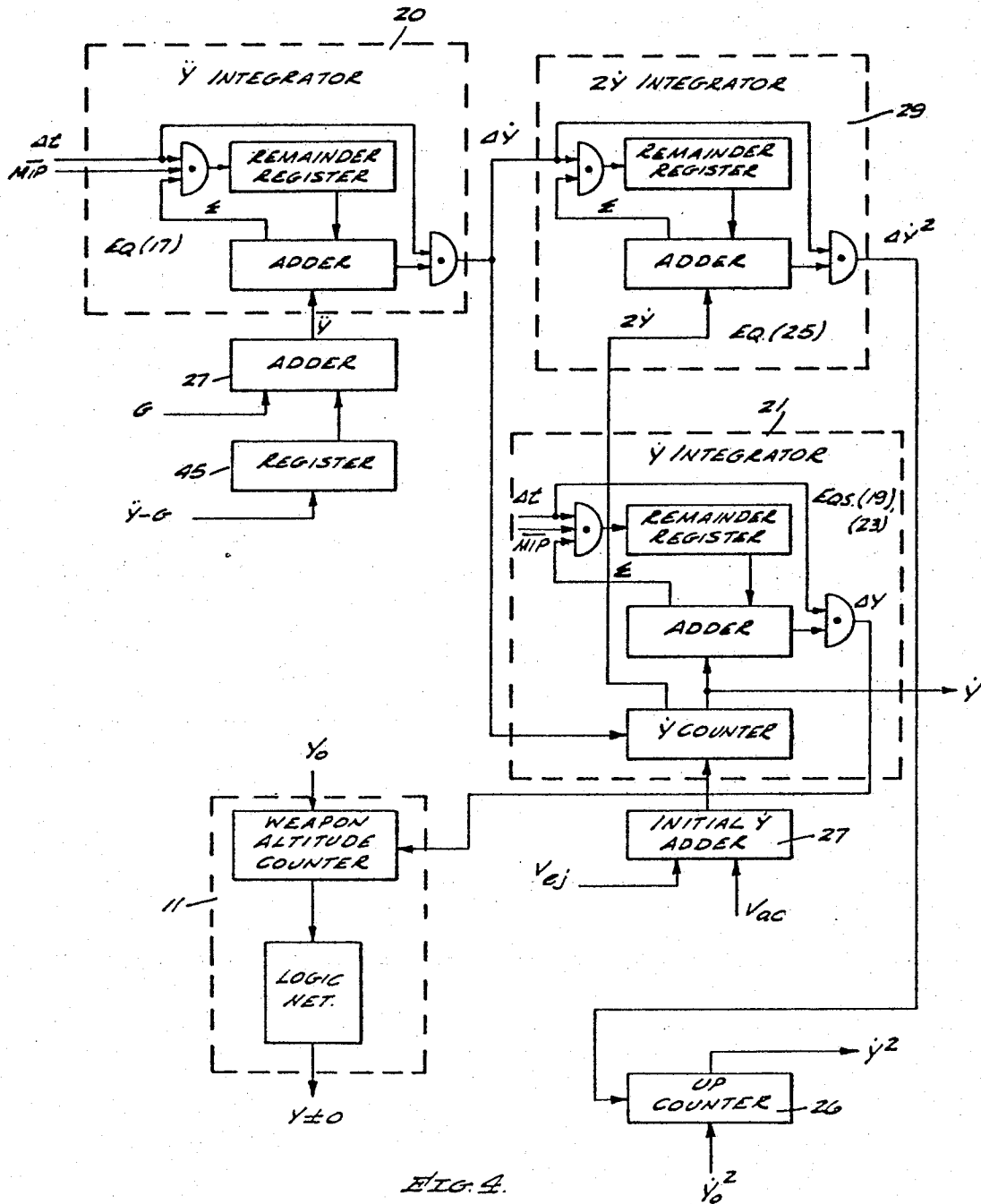


FIG. 4.

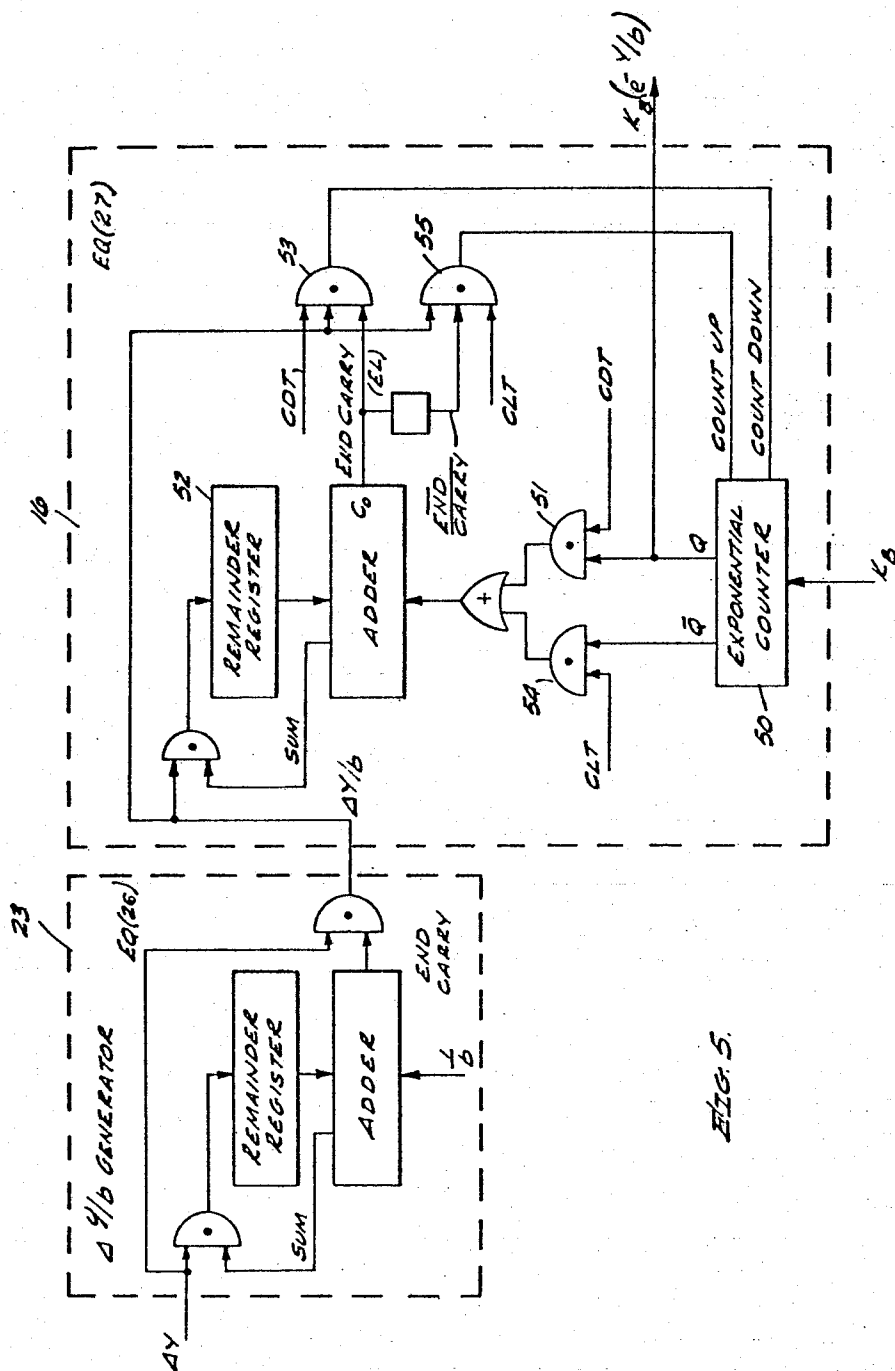
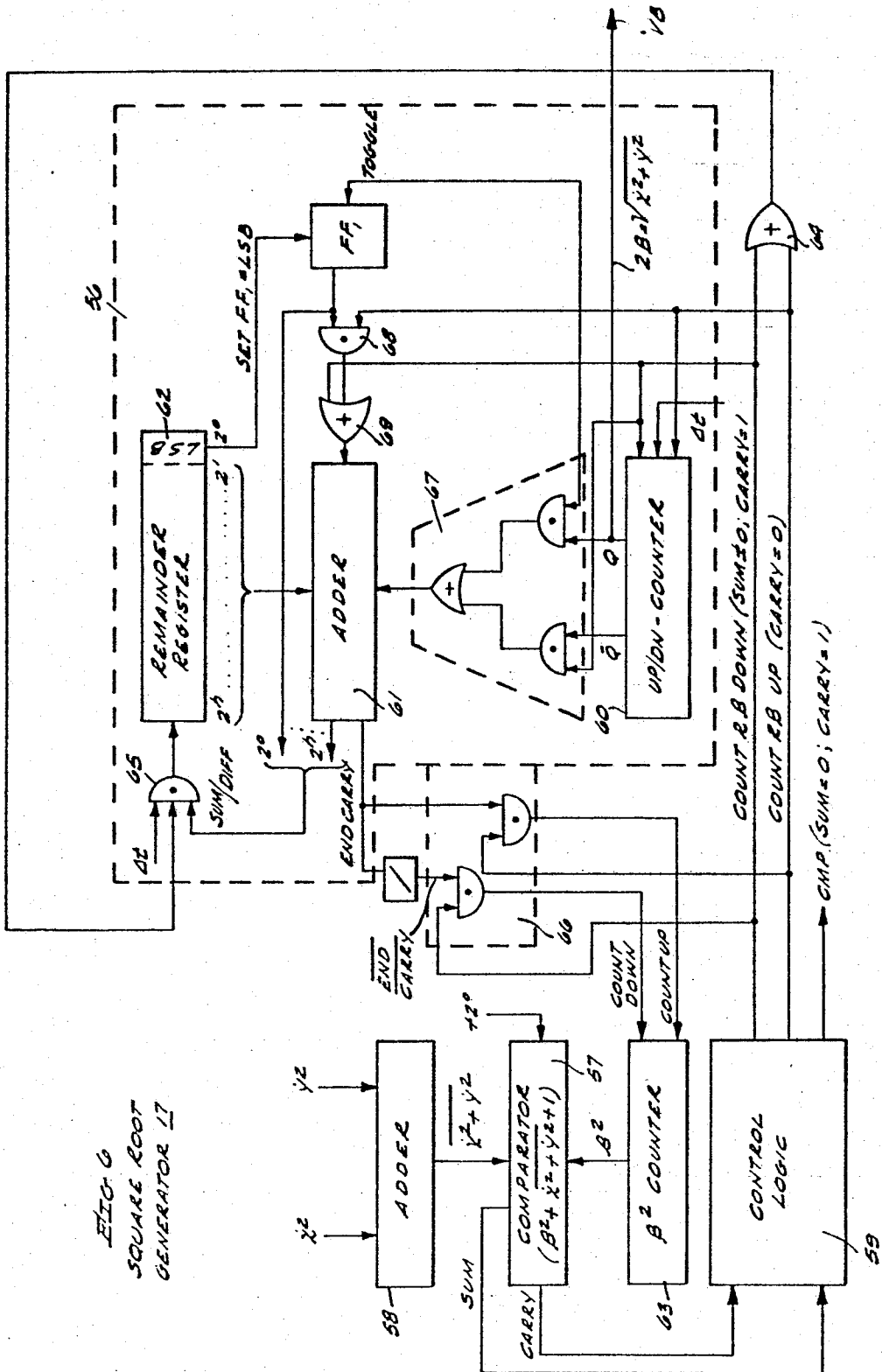


FIG. 5.

FIG. 6
SQUARE ROOT
GENERATOR 17



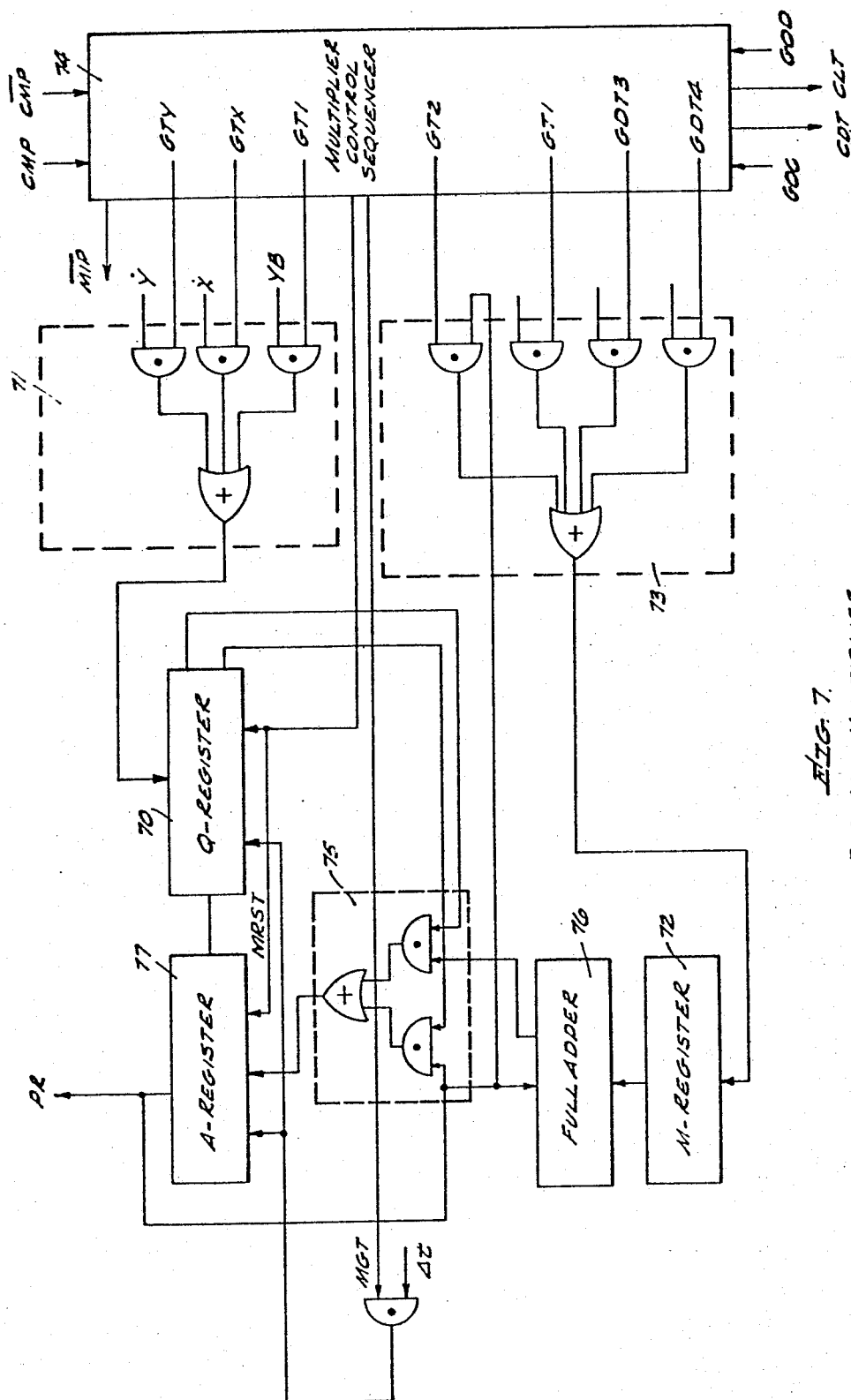


FIG. 7.
DIGITAL MULTIPLIER
SEQUENCE CONTROL 18

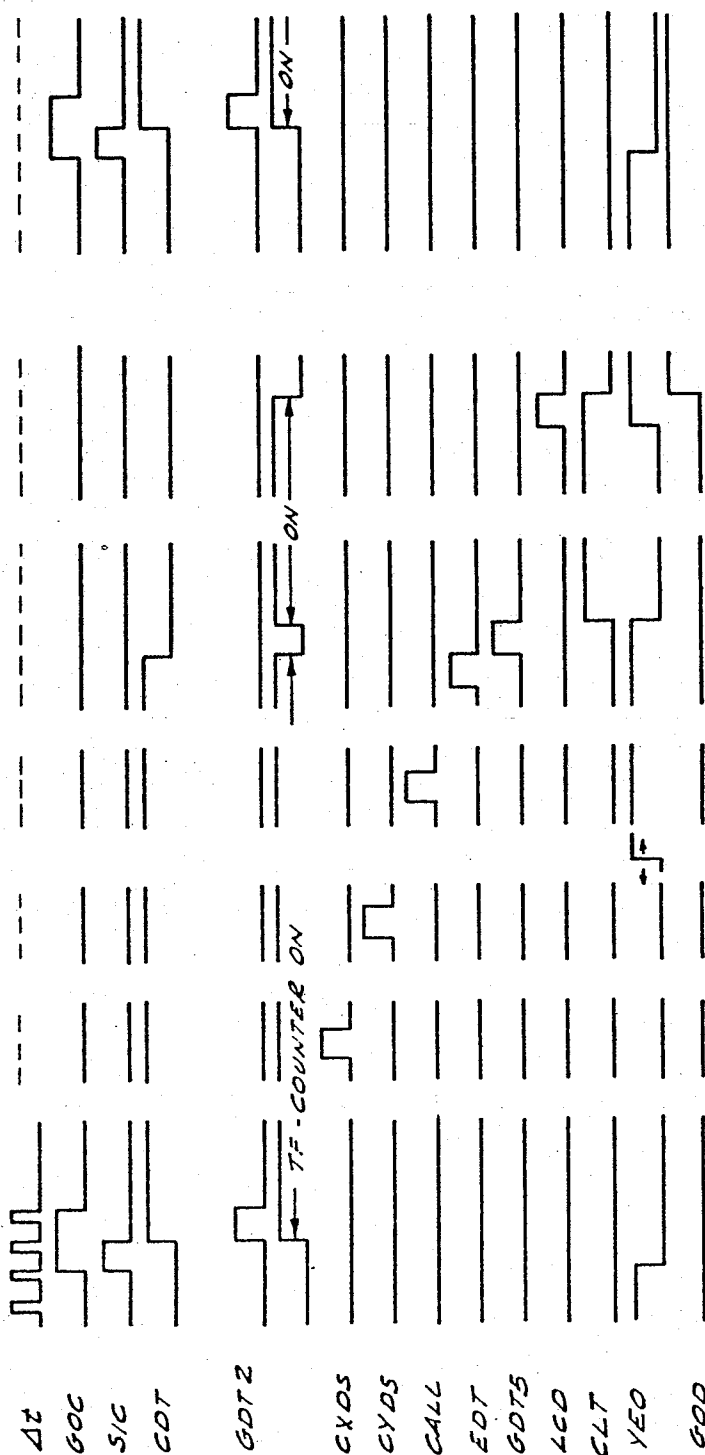


FIG. 8

BALLISTIC TRAJECTORY COMPUTER

The invention herein described was made in the course of or under a contract with the United States Air Force.

BACKGROUND OF THE INVENTION

The present invention relates to a ballistic trajectory computer, and more particularly to apparatus for computing the time of fall and range of a weapon from a known altitude above a target.

The best known method of computing time of fall and range is to approximate a solution by a binomial expansion technique forcing the known miss distance of the weapon to zero. The data is weighted upon the most probable altitude and velocity of the approach. Other precision trajectory calculating techniques require large digital computers and excessive real time calculations.

An analog computer would provide real time calculations, and in many respects is the simplest to construct, but analog techniques will not always produce the desired degree of accuracy, and require too much space. A system capable of accurately computing time of fall and range in real time would be highly desirable for providing such data to a weapon release computer, or for training in simulated practice missions. Also, it would be very desirable to provide such a system with a minimum of equipment and to operate it without substantial data memory requirements.

SUMMARY OF THE INVENTION

In accordance with the present invention, a computer for time of fall and range of a bomb-type weapon utilizes ballistic equations which assume a cool exponential atmosphere and neglect coriolis and spin effects, and which express horizontal and vertical acceleration as simultaneous second order non-linear differential equations. The computer employs digital integrators in a multi-loop system. Digital integrators are first controlled by a sequencer to compute initial conditions for the differential equations and then controlled to compute the ballistic trajectory using initial condition values automatically in proper position by the time of transition from one mode of operation to the other. When the computed ballistic trajectory has reached the impact point, the sequencer stops, and time of fall and range are available at the output terminals of two counters.

Initial conditions are computed from initial vertical and horizontal velocities, height of aircraft relative to a standard pressure plane and a constant K_B characteristic of the weapon. The initial conditions computed include an exponential $K_B e^{-Y/B}$, vertical and horizontal velocities squared, the square root of the sum of vertical and horizontal squared, and vertical and horizontal accelerations. Only integrators and a time shared multiplier are required for calculation of these initial conditions. The ballistic trajectory computations employ as an initial input aircraft height relative to the target. Whenever the exponential, vertical, or horizontal velocity of the aircraft, or the computed ballistic velocity of the weapon changes, trajectory computations are interrupted while the corresponding vertical and horizontal acceleration is recalculated; then computations are resumed with new acceleration values.

It is therefore an object of this invention to provide a simplified system for calculating time of fall and range of a ballistic weapon.

It is a further object of this invention to provide an improved system for calculating a time of fall and range of a ballistic weapon employing an exponential function referenced to a standard pressure plane.

It is another object of this invention to provide an improved ballistic trajectory computer utilizing digital integrators in closed loops for solutions of simultaneous second order differential equations.

It is still another object of this invention to provide for the solution of initial conditions for simultaneous second order differential equations utilizing some of the same digital integrators employed in the solution of those equations.

DESCRIPTION OF THE DRAWINGS

The novel features of the invention, as well as the invention itself, both as to its organization and method of operation, will best be understood from the accompanying drawings, in which like reference characters refer to like parts, and in which:

FIG. 1 is a simplified schematic diagram of a ballistic trajectory computer in accordance with the invention;

FIG. 2 is a schematic diagram of a digital integrator showing in dotted line a symbol employed to represent digital integrators in FIG. 1;

FIGS. 3 through 7 are detailed block diagrams for further explaining the ballistic trajectory computer of FIG. 1; and

FIG. 8 is a general timing diagram of a sequencer for the system of FIG. 1.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention employs both digital computer techniques and digital integrator techniques so that it may be referred to as a digital integrator-computer hybrid technique as opposed to a digital-integrator, analog-computer hybrid technique, although considered in its broadest aspects, analog computer techniques may be utilized in place of the digital computer techniques. The present invention solves the simplest ballistic equations of motion which assume a cool exponential atmosphere and neglect coriolis acceleration and spin effects. Thus, using the geometry and physics of the problem, the following simultaneous non-linear differential equations may be derived;

$$\ddot{X} = -K_{DP} \frac{d^2}{W} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{X} \quad (1)$$

$$\ddot{Y} = -K_{DP} \frac{d^2}{W} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{Y} + G \quad (2)$$

where:

$\ddot{X}$ = acceleration of the weapon in the horizontal (X) direction.

$\ddot{Y}$ = acceleration of the weapon in the vertical (Y) direction.

$\dot{X}$ = ground velocity of the weapon.

$\dot{Y}$ = vertical velocity of the weapon.

G = gravitational constant (32.174 feet/sec.²).

d = diameter of the weapon (feet).

W = weight of the weapon in pounds.

K_D = drag coefficient of the weapon (function of Mach number only).

ρ = air density (approximated by $\rho = \rho_0 e^{-Y/b}$).

ρ_0 = air density (lbs/ft³) at a standard data plane.

Y = instantaneous weapon height with respect to the standard data plane.

b = best fit constant over domain of interest (e.g. 0 to 15,000 feet).

By substituting for ρ in Equations (1) and (2) the value $\rho_0 e^{-Y/b}$, the following equations are derived.

$$\ddot{X} = -K_D \rho \frac{d^2}{W} e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{X} \quad (3)$$

$$\ddot{Y} = -K_D \rho \frac{d^2}{W} e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{Y} + G \quad (4)$$

Equations (3) and (4) may be simplified by use of a constant K_R computed in advance from the following equation:

$$K_R = K_D \rho d^2 / oW \quad (5)$$

where ρ_0 is given by standard atmospheric tables for a level chosen at a standard pressure plane below mean sea level, such as -1200 feet. This will give only an approximate value for K_R ; to obtain a more accurate value, several problems may be run for each weapon type, with the target at standard mean sea level, and the results compared with bombing tables based on a target at standard mean sea level. Upon substituting the constant given by Equation (5) in Equations (3) and (4), the trajectory equations employed in the present invention are provided in the following form:

$$\ddot{X} = -K_R e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{X} \quad (6)$$

$$\ddot{Y} = -K_R e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{Y} + G \quad (7)$$

Equations (6) and (7) can be expressed in integral form as follows:

$$X = \int_0^t \int_0^t \ddot{X} dt dt = \int_0^t \int_0^t -K_R e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{X} dt dt \quad (8)$$

$$Y = \int_0^t \int_0^t \ddot{Y} dt dt = \int_0^t \int_0^t (-K_R e^{-Y/b} \sqrt{\dot{X}^2 + \dot{Y}^2} \dot{Y} + G) dt dt \quad (9)$$

However, the present invention solves the parametric trajectory Equations (8) and (9) in the form expressed in Equations (6) and (7). Integration of variables, the most critical part of each of the Equations (6) and (7), is accomplished by digital integrator techniques while each multiplication of factors indicated is carried out using digital computer techniques. Thus, in each equation, integration is carried out by the use of digital integrators for speed, economy and accuracy, while multiplication is carried out by a single time shared multiplier for economy and accuracy.

Referring now to FIG. 1, a simplified schematic diagram of a computer is shown for calculating Equations

(6) and (7) in accordance with the present invention. The computer accepts inputs of the constant K_R , initial horizontal velocity $\dot{X}_0$, weapon vertical ejection velocity V_{EJ} and aircraft vertical velocity V_{AC} from which initial vertical velocity $\dot{Y}_0$ is determined by an adder 10. The computer also accepts the height Y of the aircraft relative to the target for solution of those equations, but first accepts height Y of the aircraft relative to a standard pressure plane to generate all initial conditions used to solve the equations. In each instance, the height Y is entered into an altitude down counter 11. In solving the equations, a time of fall (TF) counter 12 starts with a count of zero and counts up increments of time Δt from a clock generator 13 via an AND gate 14 until the altitude counter 11 has counted down from the height Y of the aircraft above the target to zero, which is when the computed weapon trajectory has reached to the impact point ($Y = 0$).

The operating sequence of the computer is controlled by internally generated timing signals as will be described with reference to FIG. 8. For the present it is sufficient to understand that a control signal CLT enables the various components of the computer to operate including, the altitude counter 11, the time of fall counter 12 and a range counter 15 (which counts up from zero to a number representing the range of the trajectory) during the time of fall determined by the counter 12. In that manner, the counters 12 and 15 provide the solution (time of fall and range) to the problem while the altitude counter 11 counts down from Y to zero, a process which takes approximately 16 to 40 milliseconds (including the time necessary to generate initial conditions) when Δt is a suitably weighted clock pulse. Once the altitude counter 11 has reached zero, operation of the range counter 15 is stopped by turning off at pulses to the integrators from the clock generator 13. That control is represented in FIG. 1 as $Y = 0$, the same signal which enables the AND gate 14 until the counter 11 has reached zero.

The two computed quantities, weapon ground range X and weapon time of fall T may be used for training in simulated bombing runs, for example, or to correct the range and bearing to the target for wind velocity in an actual bombing run. To accomplish that, the computed weapon ground range is compared with wind-corrected range by a weapon release computer (not shown), and when the two are equal, the weapon release computer generates a command to drop the weapon.

Aircraft height Y is referenced to a standard atmospheric pressure plane while generating initial conditions. That plane is purposely chosen to be substantially below standard mean sea level, such as 1200 feet below. Therefore, zero altitude for the weapon release computer (not shown) is equivalent to 1200 feet below standard mean sea level, and the height of the aircraft relative to this standard pressure plane is required as an input to the altitude counter 11 in order to initially generate the exponential $K_R e^{-Y/b}$. That exponential is sometimes referred to hereinafter simply as "the exponential." Thus the exponential represents the atmospheric density at any aircraft altitude Y entered into the computer 11 to determine range and time of fall to the target.

The constant K_R of the exponential is entered into a digital integrator 16 having its differential output

fed back to generate the exponential. The exponential generator 16 is employed with a square root generator 17 and a multiplier 18 (which includes sequence control logic) to generate the initial conditions for solution of the Equations (6) and (7). The initial conditions include horizontal acceleration $\ddot{X}$ and vertical acceleration $\ddot{Y}$ which are stored in respective horizontal and vertical digital integrators 19 and 20. The initial conditions calculated also include ground velocity squared $\dot{X}_0^2$ and vertical velocity squared $\dot{Y}_0^2$ from which initial ballistic velocity VB_0 is calculated by the square root generator 17 as the square root of the sum of vertical velocity squared and horizontal velocity squared. It should be noted that a mode control signal CDT from the sequence control portion of the multiplier 18 permits the computer to calculate all initial conditions. When they have been computed, the mode control signal CDT is terminated and a mode control signal CLT is automatically generated to allow the computer to proceed with solution of the Equations (6) and (7). At that time, height Y relative to the target is entered into the counter 11.

To summarize the general description thus far, before initial conditions are computed, the following inputs are entered (as by a weapon release computer not shown): (1) initial vertical velocity $\dot{Y}_0$; (2) initial horizontal velocity $\dot{X}_0$; (3) height of aircraft Y relative to a standard pressure plane; and (4) a constant K_B . The first two inputs are entered into respective integrators 21 and 22. Still another input is required after the initial condition computation, namely the height of the aircraft Y relative to the target. That input may be received by the computer at the same time other inputs are entered and stored in a temporary register (not shown) for transfer to the counter 11 at the end of the CDT mode of operation.

Once the inputs have been entered, the initial computation begins in response to the mode control signal CDT. As the weapon altitude counter 11 counts down, the exponential is generated by an integrator 23 and the integrator 16. When the weapon altitude counter reaches zero, an AND gate 24 terminates the accumulation of Y increments (ΔY) from the integrator 21 in the integrator 23. The integrator 16 will then contain the exponential required for the aircraft height Y relative to the standard pressure plane.

While the exponential is being generated, the multiplier 18 receives the initial horizontal velocity $\dot{X}_0$ from the integrator 22 and multiplies that value by itself. The product $\dot{X}_0^2$ is then transferred to a down-counter 25 in order to provide to the square root generator 17 the square of initial horizontal velocity. It should be noted that horizontal velocity decreases with time so that during the solution of Equations (6) and (7), the counter 25 will count down from the initial horizontal velocity squared. The initial vertical velocity $\dot{Y}_0$ is then transferred to the multiplier 18 where it is squared, and the product is transferred to an upcounter 26. From the initial values entered into the counter 25 and 26, the initial ballistic velocity VB_0 is computed by the square root generator 17. These computed values are then employed to calculate initial horizontal and vertical acceleration in accordance with the following equations:

$$\begin{aligned}\ddot{X}_0 &= VB_0 K_B e^{-Y/Y_0} (\dot{X}_0) \\ \ddot{Y}_0 &= -VB_0 K_B e^{-Y/Y_0} (\dot{Y}_0) + (G)\end{aligned}$$

(10)

These equations are computed by the multiplier 18 by first multiplying the exponential with the initial ballistic velocity VB_0 and storing the product in a multiplicand register. Horizontal velocity $\dot{X}_0$ is then transferred into the multiplier register for a second multiplication, after which the product is transferred to the integrator 19 as the horizontal acceleration $\ddot{X}$. The multiplicand register is still storing the product of the exponential and ballistic velocity so that to provide the product $\ddot{Y}-G$, it is only necessary to transfer vertical velocity $\dot{Y}$ into the multiplier register for a third multiplication. The product $\ddot{Y}-G$ is entered into the integrator 20 through an adder 27 where acceleration G due to gravity is added. Thus, the actual value stored in the integrator 2 is true vertical acceleration $\ddot{Y}$, thereby taking into account acceleration due to gravity as well as vertical velocity of the aircraft V_{ac} and vertical ejection velocity V_{ej} of the weapon relative to the aircraft, both of which are added by the adder 10 to provide the initial vertical velocity $\dot{Y}_0$. This completes the generation of all initial conditions and the computer is ready for the next mode of operation. However, before the mode control signal CLT starts solution of the Equations (6) and (7), the height Y of the aircraft relative to the target is entered in the counter 11.

When the ballistic trajectory computation begins, the exponential integrator 16 still contains the exponential generated during the initial condition computation mode CDT. As the altitude counter 11 counts down, the AND gate 24 transmits increments of altitude ΔY to the integrator 23 which, in cooperation with the integrator 16, computes the value of the exponential along the trajectory path. Simultaneously, the AND gate 14 transmits increments of time Δt to the counter 12 to determine the time it will take the weapon to fall from the initial height of the aircraft to the target, and the counter 15 counts increments of horizontal distance ΔX to determine the distance the weapon will cover over the ground while it is falling from the initial altitude of the aircraft to the target.

It should be noted that the products required by Equations (6) and (7) are generated by the multiplier 18 during the ballistic trajectory computation mode CLT (which is during the period the counter 11 is counting down from the height of the aircraft relative to the target to zero) and that while these products are being generated, one at a time, operation of the integrators is suspended.

Because the integrators are normally operating in response to clock pulses representing increments of time Δt (except while products are being generated by the multiplier 18), the solution of the Equations (6) and (7) is said to be carried out in "real time," meaning that each increment of time Δt represents a scaled increment of actual time such that the output of the counter 12 represents the actual time of flight when multiplied by an appropriate conversion factor, even though the altitude counter 11 requires only milliseconds to count down from the altitude of the aircraft relative to the target. Therefore "real time" computation here is not to be confused with "actual time" computation where each increment of time Δt is equal to exactly the actual time lapsing. Here Δt is a weighted clock equal to, for example, 2 milliseconds occurring at a frequency of 2 MHz.

Before proceeding with a more detailed description of the computer illustrated in FIG. 1 with reference to FIGS. 2 through 8, it should be noted that the ballistic trajectory computation continues without interruption until the altitude counter 11 counts down to zero, except when any of the conditions including the exponential and the square of horizontal or vertical velocity changes, in which case either the horizontal or vertical acceleration, or both, must be recalculated by the multiplier 18. All the integrators stop during these recalculations by the multiplier 18, just as during any other multiplication operation. That is readily accomplished by generating a signal MIP in the sequence control section of the multiplier whenever multiplication is in progress, and using the signal MIP for inhibiting Δt pulses to the integrators. When the multiplication has been completed, computations resume with recalculated values for horizontal and vertical acceleration in the integrators 19 and 20.

If only the horizontal velocity changes, the new value is transferred to the multiplier register by the sequence control section of the multiplier in response to an overflow from the integrator 22 indicating horizontal velocity has changed, and the new horizontal acceleration is generated as a product by the multiplier 18. At the conclusion of the multiplication, a new value of horizontal acceleration is entered into the integrator 19. A similar sequence is followed by vertical velocity changes, and the product $\dot{Y}-G$ is added to G . The sum is then stored in the integrator 20.

Proceeding now with a more detailed description of the present invention, the basic operation of a digital integrator will be described with references to FIG. 2 which shows in a dotted line 30 the basic components of a "digital integrator." the form of the dotted line is used in FIG. 1 to represent digital integrators having the same basic components. However, it should be understood that, strictly speaking, an integrator is not complete without an overflow accumulator which is sometimes provided by a counter in a following integrator, and sometimes by a special accumulator (counter). Accordingly, the basic components contained in the dotted line 30 are more often properly referred to as a digital differential analyzer or generator. However, in view of the fact that the components are used here as integrators when coupled to an accumulator, they are referred to as digital integrators, the accumulating function always being inferred since that function is always provided by a following component.

The main function of the integrator is the evaluation of the following equation expressed in integral form:

$$Z = \int_{X_0}^{X_1} Y(X) dX \quad (11)$$

The manner in which that evaluation is carried out is better understood from its expression in the following differential form:

$$dZ = Y(X) dX \quad (12)$$

The variable Z is expressed in integral form by the following equation:

$$Z = Z_0 + \int_{Z_0}^Z dZ \quad (13)$$

In the digital integrator the differentials dX , dY and dZ are replaced by finite increments ΔX , ΔY and ΔZ such that Equations (11) and (12) may be expressed as follows:

$$\Delta Z = Y \Delta X \quad (14)$$

$$Z = \sum_{i=1}^N Y_i \Delta X_i + Y_0 X_0 \quad (15)$$

The finer the resolution in ΔX , the closer the sum of Equation (15) approaches the true value of Equation (11).

The basic components of a digital integrator implement Equation (14). A Y -register 31 contains the current value of the integrand (Y) which may be a constant, in which case the Y -register may be a static register, and the ΔY input shown is equal to zero, or it may be a variable in which case the Y -register is a counter and the ΔY input is a binary signal equal to one or zero. Equation (15) is then implemented by a counter 32 adapted to count the integrator overflow generated as ΔZ increments.

For every step in the X direction, the integrator receives a ΔX increment and, in response thereto, adds the current value of the Y -register to the contents of an R -register 33, via a parallel adder 34. The sum of the addition process is returned to the R -register via a circuit shown as a single gate 35, but which actually comprises a bank of gates enabled by a ΔX increment to enter into the R -register the output of the adder 34 in parallel. If the R -register 33 is implemented with a bank of D -type flip-flops, the gating function may be carried out by applying the ΔX increment to each flip-flop as its clock pulse (CP). The inputs of the flip-flops from the adder 34 at the time a ΔX increment occurs completely determines the binary number stored in the R -register.

During the process of integration, the ΔX increments keep adding the contents of the Y -register to the contents of the R -register, and the R -register will overflow via an AND gate 36 as soon as its capacity is exceeded. For example, if the R -register is a 4-bit register storing the binary number 1111, when the binary number 0110 in the Y -register is added, the sum stored in the R -register is 0101. The carry from the most significant bit position of the adder is the overflow transmitted to the counter 32, or some other accumulator, such as another digital integrator, as the increment ΔZ .

As the overflow increments are accumulated, their sum at any given time may be combined with the contents of the R -register by assuming they are the successively higher orders of the binary number stored in the R -register, i.e., by considering the flip-flops of the counter 32 as an extension of the R -register 33. The sum of the integration process is therefore combined in the R -register 33 and the counter 32, or a Y -register of another digital integrator implemented as a counter.

The frequency of the ΔZ increments will depend on the frequency of the ΔX increments, and the current value of the integrand Y . If Y is very large, an overflow

will occur practically every time a ΔX increment occurs. If ΔX increments occur at a slow rate, ΔZ increments will also occur at a slow rate. If the integrand Y is equal to zero, there will be no overflow regardless of the frequency of the ΔX increment. This condition satisfies Equation (12).

The accuracy of the integration given by Equation (13), but implemented by Equation (15), depends only on the accuracy of the integrand Y and the resolution in ΔX . The higher the degree of accuracy required, the greater the number must be of binary digits for the integrand Y . The adder 34 and the R-register 33 are then provided with capacity for the same number of binary digits. The size of the counter 32, on the other hand, depends on the maximum number of overflows expected during the period of integration.

The generation of various mathematical functions with digital integrators necessary to implement Equations (6) and (7) in accordance with the present invention will now be described with reference to FIGS. 3 to 6 assuming the following identities are known:

$$X = \int_0^t (V_X = \dot{X}) dt = \int_0^t \int_0^t (a_X = \ddot{X}) dt dt$$

$$Y = \int_0^t (V_Y = \dot{Y}) dt = \int_0^t \int_0^t (a_Y = \ddot{Y}) dt dt$$

Using these identities in integral form, a series of equations in differential form are implemented for the present invention, as follows:

$$\dot{X}_0 \Delta t = \Delta \dot{X}_1$$

(16)

$$\dot{Y}_0 \Delta t = \Delta \dot{Y}_0$$

(17)

$$\dot{X}_0 \Delta t = \Delta X_0$$

(18)

$$\dot{Y}_0 \Delta t = \Delta Y_0$$

(19)

$$\dot{X}_1 = \dot{X}_0 - \Sigma \Delta \dot{X}_0$$

(20)

$$\dot{Y}_1 = \dot{Y}_0 + \Sigma \Delta \dot{Y}_0$$

(21)

$$X_1 = X_0 + \Sigma \Delta X_0$$

(22)

$$Y_1 = Y_0 - \Sigma \Delta Y_0$$

(23)

$$-\Delta \dot{X}_0^2 = (-2 \dot{X}_0 + \Delta \dot{X}_0) \Delta \dot{X}_0$$

(24)

$$\Delta \dot{Y}_0^2 = (2 \dot{Y}_0 + \Delta \dot{Y}_0) \Delta \dot{Y}_0$$

(25)

Generation of an exponential and of a constant multiplied by a variable may be in accordance with the following equations:

$$1/b \Delta Y_0 = \Delta Y_0 / b$$

(26)

$$K_B \Delta e^{-Y_0/b} = -K_B / b (e^{-Y_0/b}) \Delta Y_0$$

(27)

FIG. 3 is a block diagram of a preferred implementation of the horizontal function of Equations (16), (18), and (24) generated by respective digital integrators 19, 22 and 28 (as indicated by legends within the dotted lines of each for convenience). Equation (22) is then implemented by the range counter 15 which is initially set to zero ($X_0 = 0$), and the current value of $\dot{X}^2$ is provided by adding $-\Delta \dot{X}^2$ to the initial value of $\dot{X}_0^2$ in the counter 25 (implemented by utilizing a counter which counts down from X_0^2 in response to the overflow from the integrator 28).

In the integrator 19, a register 40, which corresponds to the Y -register 31 of FIG. 2, receives as the integrand the calculated initial value of horizontal acceleration $\ddot{X}$. That value remains constant until it is recalculated. Therefore, there is no input to the register 40 which corresponds to ΔY in FIG. 2. In the integrator 22, on the other hand, a corresponding register 41 must receive an initial value of horizontal velocity as a variable. Therefore, the register 41 is a down-counter able to count down the ΔX increments from the integrator 19. The counter 41 provides the current value of horizontal velocity $\dot{X}$ in accordance with Equation (20) as an output to the multiplier 18, and twice that value to the integrator 28 as the integrand $-2\dot{X}$. The negative sign is not attached; instead it is implied by always subtracting $\Delta \dot{X}^2$ increments from the initial value $\dot{X}_0^2$ in the counter 25, i.e., by counting down overflow pulses from the integrator 28. The multiplier "2" is introduced by connecting to the integrator 28 as the integrand all but the least significant digit of the counter 41, thereby effectively multiplying by shifting up the integrand one binary position.

FIG. 4 similarly shows a preferred implementation of the vertical functions given by Equations (17), (19), (21) and (25). The current value of $\dot{Y}^2$ is provided by adding $\Delta \dot{Y}^2$ in the counter 26. Equation (23) is implemented by the altitude counter 11 which includes a logic network for determining when Y is not equal to zero. When the logic network decodes zero in the counter, the enabling voltage signal to gates 14 and 24 of FIG. 1 is removed and the current operation mode CDT or CLT is terminated.

The integrator 20 receives as the integrand the initial vertical acceleration value $\ddot{Y}$ which remains constant until recalculated by the multiplier 18 (FIG. 1). However, the multiplier calculates only the value $\ddot{Y}-G$; the adder 27 then adds the constant G . Because a constant

is being added, it is preferred to place the adder 27 at the output of the register for the integrator 20. Accordingly, a register 45 is placed at the input of the adder 27, as shown. Thus, the register 45 and adder 27 combine their functions to store the integrand Y for the integrator 20.

The integrators 21 and 29 correspond to the integrators 22 and 28, and are therefore implemented in the same way. The only difference in generating $\dot{Y}^2$ is that while $\dot{X}^2$ is decreasing from an initial value, due to the geometry and physics of the problem, $\dot{Y}^2$ is increasing from an initial value. That difference is taken into account by providing the counter 26 as an up-counter, i.e., as a counter incremented by overflow increments $\Delta\dot{Y}^2$ from the integrator 29, and not as a down-counter decremented as in the counter 25 (FIG. 3).

Referring now to FIG. 5, the exponential $K_B e^{-Y/b}$ is computed by integrators 23 and 16 in accordance with Equations (26) and (27), respectively. Inputs K_B and $1/b$ are constants, and Y is an incremental input ΔY . The integrator 23 operates as a multiplier to carry out the function of Equation (26). Each time an increment ΔY is received, $1/b$ is added to the contents of the remainder register corresponding to register 33 of FIG. 2. The overflow represents the product $\Delta Y/b$. The overflow is the incremental input to the digital integrator 16, which must be capable of generating both an increasing and a decreasing function.

To generate a decreasing function starting from $Y = 0$, K_B is set into an up-down counter 50 and a mode control signal CDT is applied to a control gate 51. K_B is the value of $K_B e^{-Y/b}$ when $Y = 0$. Each time a $\Delta Y/b$ input is received, the contents of the counter 50 is added to the contents of a remainder register 52, and the sum is stored in the remainder register. Whenever an overflow occurs from the adder as an end carry, the counter 50 is decremented (counted down) via a gate 53. The instantaneous value of $K_B e^{-Y/b}$ is obtained from the counter 50. Gate 51 is shown as a single gate, but it is to be understood to represent a bank of gates, one for each binary order.

To generate an increasing function, which is equivalent to a negative $\Delta Y/b$, the mode control signal CDT is removed and a mode control signal CLT is applied to a gate 54. The integration process now reverses. Each time a $\Delta Y/b$ input is received, the contents of the counter 50 is subtracted from the contents of the remainder register 52, and the difference is stored in the remainder register. Whenever an overflow does not occur from the adder as an end carry, the counter 50 is incremented (counted up) via a gate 55. Gate 54 is shown as a single gate, but it is to be understood that it represents a bank of gates and means for generating a carry into the least significant bit position to form the 2's complement in order to subtract by adding the 2's complement of the integrand in the counter 50.

In greater detail, the output of the $\Delta Y/b$ integrator 23 is the variable input to the exponential integrator 16. Initially the exponential counter 50 is reset to zero and then set to K_B , the weapon constant which is equal to $K_B e^{-Y/b}$ when $Y = 0$. During computation of initial conditions (mode CDT) the integrator generates $K_B e^{-Y/b}$, where Y is the altitude above a standard pressure plane. Each time an increment $\Delta Y/b$ is received, the contents of the counter 50 is added to the contents of the remainder register 52 and the sum is returned to the remainder register. Whenever an overflow occurs from

the adder, the exponential counter 50 is counted down. When initial conditions have been computed, the counter 50 contains $K_B e^{-Y/b}$ where Y is the aircraft altitude above the standard pressure plane.

When solution time (mode CLT) starts, the counter 50 still contains $K_B e^{-Y/b}$. The integration process now reverses so that the counter 50 will contain the correct value of $K_B e^{-Y/b}$ as the weapon falls. Each time an increment $\Delta Y/b$ is received, the counter 50 counts up if an overflow does not occur from the adder. The contents of the counter 50 are subtracted from the contents of the remainder register 52 and the difference is stored in the remainder register.

The mechanization of the square root generator 17 of FIG. 1 will now be described with reference to FIG. 6. Its function is to generate the square root of the sum of two binary numbers ($\dot{X}^2 + \dot{Y}^2$) for an increasing or decreasing function. An output signal (CMP) is also generated that indicates when a square root solution has been achieved.

The square root generator consists primarily of a closed loop digital integrator 56 and a comparator 57 acting like a digital servo. When either $\dot{X}^2$ or $\dot{Y}^2$ increases or decreases, the closed loop is no longer in a state of equilibrium, and the integrator 56 will operate until a new square root has been computed. The two inputs ($\dot{X}^2$ and $\dot{Y}^2$) are summed in a parallel adder 58. This sum is compared with the output β^2 of the integrator. When $\dot{X}^2 + \dot{Y}^2 = \beta^2$, control logic 59 stops the integrator, and the square root is contained as the integrand in a counter 60, as signified by a signal CMP from the control logic 59 being turned on.

If ($\dot{X}^2 = \dot{Y}^2$) is greater than β^2 , the output from the control logic will generate a count-up control signal for a counter 60. This count-up control signal will initiate the following sequence:

1. Count the counter 60 up two increments ($2\Delta\beta$) on each clock pulse Δt ;
2. Add $\Delta\beta$ to the contents of a remainder register 62 by injecting an input carry into the least significant bit (LSB) of an adder 61 when the least significant bit of the register 62 is true, and set that least significant bit to zero; otherwise set the least significant bit to one.
3. Add 2β to the contents of a remainder register 62, and store the sum in the remainder register on each clock pulse;
4. Accumulate end carries in a β^2 counter 36 by counting up; and
5. Compare the new value of β^2 with $\dot{X}^2 + \dot{Y}^2$. When they are equal, the count-up command will go off, the integration process will stop, and the signal CMP will be true.

If $\dot{X}^2 + \dot{Y}^2$ is less than β^2 , the control logic 59 will generate a count-down control signal which will initiate the following sequence:

1. Count the counter 60 down two increments ($2\Delta\beta$) on each clock pulse Δt ;
2. Inject an input carry into the adder in its least significant bit position (LSB);
3. Add -2β to the contents of the remainder register 62 on each clock pulse; and
4. Accumulate end carries in the β^2 counter 63 by counting down; and
5. Compare the new value β^2 with $\dot{X}^2 = \dot{Y}^2$. When they are equal, the count-down control signal will go off, the integration process will stop, and the signal CMP will be true. The comparison by the comparator

57 is obtained by adding the 1's complement to $X^2 + Y^2$ in a full adder with its carry input forced to 1, and observing the sum and end carry outputs. If the sum is not zero and there is an end carry, the control logic 59 generates a count-down control signal, and if there is not a carry, regardless of the values of the sum, the control logic generates a count-up control signal; otherwise, neither control signal is on, indicating the special condition of equality which occurs when a signal is generated and the sum is zero. Static logic networks comprising conventional gates for implementing Boolean logic equations may be employed to implement these comparator functions operating on the outputs of a conventional parallel full adder.

The integrand β formed in the counter 60 is offset one binary position with respect to the remainder register 62, thereby providing the product 2β for the integrator 56 while acting like a digital servo. When $62^2 = X^2 + Y^2$, i.e., when a square root solution is achieved and CMP is true, the content of the counter 60 is read into the multiplier 18 as β by effectively shifting 2β down one binary position to constitute the current value for velocity VB.

An exemplary mechanization of the foregoing algorithms is illustrated in FIG. 6. When the control logic 59 determines that $X^2 + Y^2$ is greater than β^2 , the signal CMP is false and a "count up" signal is transmitted to the counter 60, and through an OR gate 64 to gates for storing the sum of the contents of the counter 60 and the register 62 in the register 62. The transfer is parallel through the gates represented by an AND gate 65. At the same time, one of a pair of AND gates 66 is enabled by the "count up" signal to cause an end carry from the adder 61 to increment the β^2 counter 63. The value β^2 is thus generated in the accumulator (counter 63) of the digital integrator 56 in a manner similar to the generation of Y in the counter 26 functioning as the accumulator for the integrator 29.

It should be noted that while the integrator 56 is acting like a digital servo under control of the count-up signal, the value $2\beta + \Delta\beta$ added to the contents of the register 62. The value 2β is added directly through controlled gates 67 (shown for only one order, but understood to be duplicated for each order of the counter 60). The value $\Delta\beta$ is then added by injecting a carry into the least significant bit position of the adder 61 via an AND gate 68 when the least significant bit copied into a flip-flop FF1 is true. At the same time the flip-flop FF1 is set false. When the least significant bit of the register 62 is false, a carry is not injected into the least significant bit of the adder 61, but the flip-flop FF1 is set true. In that manner the flip-flop FF1 behaves as a binary counter in response to a count up signal in that it always changes state at each clock pulse Δt . In practice, the flip-flop FF1 may actually be the least significant bit flip-flop of the register 62, but implemented as a binary counter, to avoid having to copy its contents into an external flip-flop as shown for clarity in the explanation of the operation of adding $2\beta + \Delta\beta$ to the contents of the register 62.

When $(X^2 + Y^2)$ is less than β^2 , it is necessary to decrease β^2 by adding the value $-2\beta + \Delta\beta$. That is done by adding the 1's complement of the contents of the counter 60 and injecting a 1 in the least significant bit position of the adder 61, thereby subtracting 2β by adding the 2's complement of 2β . At the same time, $\Delta\beta$ is added. Since $2\Delta\beta$ is thus always to be added when the

"count down" signal from the control logic 59 is on, a carry is always injected through an OR gate 69 in response to the "count down" signal instead of conditionally in response to the "count up" signal. The result is that $2\beta + 2\Delta\beta$ is added to achieve the objective of adding $-2\beta + \Delta\beta$ in response to a count down signal. To complete the generation of a decreasing β^2 , the gates 66 respond to a "count down" control signal to decrement the counter 63 when end carry is not present, i.e., in response to an $\overline{\text{END CARRY}}$ signal in a manner similar to generation of a decreasing X^2 in the counter 25 by the integrator 28.

Whereas all sections of the computer illustrated in FIG. 1 and described thus far with reference to FIGS. 3 to 6 are implemented with digital integrators, including the square root generator, the multiplier section 18 is preferably implemented as a conventional, time-shared, digital multiplier. The square root generator could also be readily implemented with digital techniques, such as the conventional non-restoring square root technique, but since it is used only for computing ballistic velocity VB, and for nothing else, it is more economical of time and hardware to use the digital servo technique shown.

There are many types of multipliers which may be employed, such as the simultaneous (parallel) type to which steady-state signals representing the multiplicand and multiplier are simultaneously applied to the input lines. After transients have disappeared, signals representing the product appear on output lines as long as the input signals are maintained. Then it is only necessary to program transfer of multipliers and multiplicands to respective registers to obtain the various products required. However, the exemplary embodiment illustrated in FIG. 1 contemplates use of the accumulation (shift-and-add) type as illustrated in FIG. 7. The multipliers VB, X and Y can be programmed into a Q-register by a network of gates 71 in response to control signals, GT1, GTX and GTY, respectively. The exponential $K_a e^{-Y/b}$, the product PR, X and Y can be programmed into an M-register 72 as multiplicands by a network of gates 73 in response to respective control signals.

All of the control signals are generated at the appropriate times by a multiplier control sequencer 74. Because the multiplier is time shared during calculation of initial conditions and the solution of Equations (6) and (7), which is at all times that the computer of FIG. 1 is in use, the sequencer 74 includes all the computer sequence control logic, including control of the basic modes CDT and CLT for

The sequencer 74 receives a start signal GpC from a cycle control system (not shown), such as a weapon release computer, to begin a new computation and in response thereto generates a mode control signal CDT to calculate initial conditions. Following that it automatically generates a mode control signal CLT for solution of the Equations (6) and (7) until the altitude counter 11 again counts down to zero, at which time a signal GOD is transmitted to the cycle control system signifying that computation is complete.

A more complete description of the computer sequencing will be presented hereinafter; for the present, it is sufficient to understand that the appropriate signals are generated as needed to gate multipliers and multiplicands into the multiplier. In that regard, the gates illustrated in FIG. 7 are for one binary digit only; there

are as many sets of gates 71 and 73 as there are binary digits in the multipliers and multiplicands, and as many sets of multiplier gates 75 as there are orders in an adder 76.

Initially an A-register 77 and the Q-register 70 are reset to zero by a control signal MRST from the sequencer 74. The appropriate numbers are then set into the Q-register and the M-register. Following that, the multiplication takes place while a control signal MGT added to the clock signal Δt is true for as many clock periods as there are binary digits in the multiplier since it requires one clock to complete one multiplication sequence for each bit of the multiplier.

If the least significant bit (LSB) of the Q-register is true, the following operations occur on the negative-going edge of each clock pulse while MGT is true: (1) the content of the M-register is added to the content of the A-register; (2) the sum is transferred into the A-register, but shifted down one bit position with the LSB entered into the most significant bit (MSB) of the Q-register; and (3) the content of the Q-register is shifted down one bit position in the Q-register. A new multiplier bit is thus transferred into the LSB position of the Q-register and this new bit controls the sequence on the next clock pulse. If the LSB of the Q-register is false, the following operations occur on the negative-going edge of each clock pulse while MGT is true: (1) the content of the A-register is shifted down one bit position; (2) the LSB of the A-register is shifted into the MSB of the Q-register; and (3) the content of the Q-register is shifted down one bit position in the Q-register. When the multiplication sequence has been completed the signal MGT is set false, the Q-register contains the least significant half of the product, and the A-register contains the most significant half of the product.

As noted hereinbefore, the multiplier control sequencer 74 shown in FIG. 7 as part of the multiplier control and digital multiplier 18 of FIG. 1 includes logic for the control of the entire computer using the time of fall counter 12 shown in FIG. 2 as the basic sequence control timer operating in response to clock pulse (Δt) from the clock generator 13. Thus, the sequencer 74 contains all the timing and control gates for the computer, and for the purpose of providing sequence control is presumed to include inputs from the time of fall counter 12. In addition, digital multiplier and sequence control 18 is presumed to contain three input storage registers for storing initial input data, namely initial horizontal velocity $\dot{X}$, height Y above standard pressure plane and height Y above the target. The two latter values are both referred to as the initial height or altitude Y , but at any given time in the description of the present invention, the value Y referred to during the computation of the initial conditions is, of course, the height above the standard pressure plane, while the value Y during solution of Equations (6) and (7) is understood to be the height above the target.

The two basic modes of operation of the computer are controlled by the mutually exclusive signals CDT and CLT generated by the sequencer 74. Thus, during the CDT mode, all of the initial conditions are computer so that the computer contains all the necessary values for initial conditions before initiating computation of the ballistic trajectory equations. Thereafter, when the sequencer 74 generates the mode control signal CLT, the computer calculates the ballistic equa-

tions based upon the initial conditions stored in the computer at the end of the CDT mode.

FIG. 8 shows a general timing diagram for a complete computation cycle of the sequencer 74. The timing cycle is initiated by the input signal GOC generated by an external control system, such as a weapon release computer (not shown). This initiates an SIC signal which is distributed throughout the computer to set initial conditions which includes setting the remainder registers of integrators 16, 19 to 23, 38 and 29 to one-half full count, i.e., with a bit 1 in the most significant bit position, and a bit 0 in all other positions, in order to minimize system round off errors. The time of fall counter 12 and the range counter 15 are initially set to zero at this time SIC. The remainder register 62 of the square root generator (FIG. 6) is also set to zero initially.

The SIC signal need be present for only one clock period as shown. At the end of the SIC pulse, the CDT mode control signal is generated and transmitted to the time of fall counter 12 to turn it on so that it may function as a digital clock starting with time zero. Since the counter 12 is set to zero by the SIC signal, logic gates in the sequencer 74 may decode the output of the counter 12 to initiate and terminate the various sequence control signals at appropriate times. At the same time that the mode control signal CDT is turned on, a control signal GDT2 is turned on for one clock period to enter the initial horizontal velocity $\dot{X}$ into the integrator 22, enter the initial vertical velocity $\dot{Y}$ into the integrator 21, enter the weapon constant K_a into the integrator 16 and transfer height Y relative to the standard pressure plane into the altitude counter 11.

When the counter 12 has reached a count of 16, a control signal CXDS is generated for one clock period to initiate the computation of the square of horizontal velocity. That may be accomplished by turning the down-counter 25 on so that overflow pulses from the integrator 28 may be subtracted from the square of the initial horizontal velocity computed by the multiplier in response to a control signal MGT (not shown in FIG. 8). The signal MGT gates the multiplier on for one multiplication sequence following generation of GTX and GDT3 signals (also not shown in FIG. 8) to transfer horizontal velocity $\dot{X}$ from integrator 22 to the Q- and M- registers as shown in FIG. 7. The computation of vertical velocity squared $\dot{Y}^2$ is similarly controlled by signal CYDS generated for one clock period when the counter 12 reaches the count of 64. Following that when the counter 12 has reached a count of 6,144, a control signal CALL is generated for one clock period to reset the altitude counter 11 and initiate the computation of all initial conditions. The relatively long period required to reach the count of 6,144 is provided to allow the square root generator 17 to settle and allow the exponential integrator 16 to settle for the initial height Y . Thus, after using the multiplier to generate horizontal velocity squared and vertical velocity squared, the signal CALL turns on the square root generator 17 and the exponential generator comprising integrators 23 and 16. As the altitude counter 11 counts down, the exponential is generated and when the altitude counter 11 reaches zero, the exponential integrator 16 will contain the proper exponential value for the aircraft height relative to the standard pressure plane.

When the counter 12 has reached a count of 6,208, an EDT signal is generated for one clock period. At the

end of that clock period, the mode control signal CDT is turned off and a control signal GDT5 is turned on for one clock period to transfer the height Y above the target into the altitude counter 11 in preparation for computation of Equations (6) and (7) during the CLT mode sequence to follow. The GDT5 signal may also be employed to reset the time of fall counter 12 to zero; alternatively, a separate control signal may be generated for that purpose as the end of the CDT signal. If such a separate signal is generated, a separate pulse should also be produced to reset the counter 12 during the SIC period. In the next clock pulse, the mode control signal CLT is turned on and held on until the altitude counter 11 counts down to zero, a condition necessary for a solution of Equations (6) and (7), i.e., a condition that must be satisfied to yield the time of fall T and the range X of a particular weapon having a constant K_b ejected from an aircraft with a velocity V_e at an altitude Y relative to the target while the aircraft is travelling at a known velocity V_{ac} .

When the counter 11 has counted down to zero, a signal YEO is generated. The next clock then turns off the mode control signal CLT to complete one cycle. The sequencer 74 repeats the same cycle when another GOC signal is received. As the mode control signal CLT is turned on, the GOD signal is turned on to signal to a utilization device, such as a weapon release computer, that time of flight and range data are available. Since a complete cycle requires only 16 to 40 milliseconds, a weapon release computer can continually recycle the sequencer 74 in order to have continually updated time of flight and range data until actual release of the weapon is initiated.

The sequencer 74 includes all of the control logic necessary for all operations during the CDT and CLT modes. The details of such logic may vary depending upon particular details of implementation selected for the various sections. Accordingly, detailed sequence control logic will not be described here, particularly since one skilled in the art may readily devise the detailed control logic necessary for implementing the required operations in the proper sequence thus far described. To facilitate that, the two modes will be reviewed in somewhat more detail than is illustrated by the timing diagram of FIG. 8.

During the CDT mode, three main timing sequences occur. When the control signal CDT goes true, the counter 12 is turned on to generate the basic timing for the CDT mode. At the count of 16, a multiplication sequence is initiated to generate $\dot{X}^2$. That multiplication sequence includes generation of GDT3 and GTX signals to transfer $\dot{X}$ into the multiplier and multiplicand registers. The multiplicand is then multiplied by the multiplier during the next N clock periods as determined by a control signal MGT, where N is the number of binary digits in the multiplier. Then an XDS signal is generated which transfers the product $\dot{X}^2$ from the multiplier to the down-counter 25, thus completing the first of the three main timing sequences of the CDT mode.

When the counter 12 reaches a count of 64, the multiplication sequence is repeated, except that GDT4 and GTY transfer $\dot{Y}$ into the multiplier and multiplicand registers. This number $\dot{Y}$ is thus squared during an ensuing MGT signal and the product $\dot{Y}^2$ is transferred to the counter 26 at the end of the MGT signal by a EYDS

signal to terminate the second of the three timing sequences.

The third timing sequence occurs when the counter 12 reaches the count of 6,144. Then a signal ET1 is generated to transfer the ballistic velocity V_B computer by the square root generator 17 and the exponential computed by the integrators 23 and 16 into the multiplier 18 where they are multiplied during an ensuing MGT signal. Following that, a GT2 signal transfers the product back into the M-register as shown in FIG. 7. At the same time, another GTX signal is generated to transfer horizontal velocity $\dot{X}$ into the Q-register. A second multiplication then follows during an ensuing MGT timing signal, thereby producing initial horizontal acceleration $\ddot{X}$, as the product of velocity V_B , the exponential, and horizontal velocity $\dot{X}$. The product is transferred to the integrator 19 by a control signal EXDD produced at the end of the multiplication period. The product of velocity V_B and the exponential is still in the multiplicand register so that to generate $\dot{Y}-G$, it is only necessary for a GY2 signal to gain be generated to transfer the initial vertical velocity $\dot{Y}$ into the Q-register. Another multiplication sequence then occurs during an ensuing MGT control signal. At the end of that sequence, the new product $\dot{Y}-G$ is transferred to the adder 27 where the constant G is added to provide the integrator 20 with the vertical acceleration $\ddot{Y}$.

During the CLT mode, a multiplication sequence must occur each time there is an overflow from the integrator 19, an overflow from the integrator 20 or an overflow from the exponential generator 16 since an overflow from any one of those generators indicates that initial acceleration values must be recalculated. The same is true if there is any change in the output of the square root generator 17, as indicated by the signal CMP at the output of the comparator 57 (FIG. 6) going false.

Assuming an overflow from the generator 19, a new value must be computed for $\dot{X}$. Since the multiplier 18 still contains the product of V_B and the exponential from the CDT mode, it is only necessary to generate a GT3 signal to transfer the new $\dot{X}$ into the multiplier and initiate a new multiplication sequence under control of an MGT signal. The product is then transferred to the generator 19 by an EXDD signal. If an overflow occurs from the integrator 20, the same sequence occurs except that a GT4 signal is generated to transfer the new $\dot{Y}$ into the multiplier and after the multiplication sequence, a control signal EYDD is generated to transfer the product into the integrator 20.

If an overflow occurs from the output of the integrator 16, a new GT1 signal is generated to transfer the current exponential and ballistic velocity values into the multiplier. Following that, a multiplication sequence is initiated under control of an MGT signal after which the product is transferred back into the multiplicand register by a GT2 signal. Immediately following that a GT3 signal is generated to transfer the current horizontal velocity into the multiplier register. Another multiplication sequence then follows under control of an MGT signal and the product is transferred to the integrator 19 under control of a signal EXDD. Immediately after that a GT4 signal is generated to transfer vertical velocity $\dot{Y}$ into the multiplier register for multiplication with the content of the multiplicand reg-

ister. Following that third and last multiplication sequence following an overflow from the integrator 16, the product is transferred to the adder 27 to provide the integrator 20 with a new vertical acceleration value.

A change in the output of the comparator 57 occurs when either $\dot{X}^2$ or $\dot{Y}^2$ changes value since a comparator will then momentarily indicate that the square of the output VB is not equal to $\dot{X}^2 + \dot{Y}^2$. Immediately a signal MIP is generated signifying that a multiplication operation is in progress. Operation of all integrators is suspended (as by inhibiting clock pulses to the integrators (while the signal MIP is present. It should be noted that this same MIP signal is generated at any time a multiplication operation is in progress during the CLT mode so that all integration is suspended while new initial conditions values are being generated.

Once the MIP signal has been generated in response to a change and the output of the comparator 57 of the square root generator, no other change will occur until the square root generator has computed a new value of ballistic velocity VB the square of which is equal to the then existing value of $\dot{X}^2 + \dot{Y}^2$. Then control logic 59 will indicate that a solution has been achieved by a signal CMP. When that occurs, a GT1 signal is generated to transfer the new ballistic velocity VB and the current exponential into the multiplier. Following that, two multiplication sequences are generated to compute and transfer new $\dot{X}_0$ and $\dot{Y}_0$ values into the generators 19 and 20 just as for the timing sequence which occurs when there is an overflow from the exponential integrator 16.

The signal MIP may be generated by a flip-flop MIP set in response to either an increment $\Delta\dot{X}$, an increment $\Delta\dot{Y}$, or an increment ΔEXP , the latter from the exponent generator (digital integrator 16), at any time during the compute mode CLT. Once the flip-flop MIP is set, operation of a multiplication sequence counter is initiated to generate the control signals required, such as the signal MGT. Separate storage flip-flops are provided to store the received increments $\Delta\dot{X}$, $\Delta\dot{Y}$ or ΔEXP in order to select the proper multiplier sequence and the appropriate one of the product transfer control signals (EXDD and EYDD) at the end of the multiplication sequence. Following the indicated product transfer, or transfers, the storage flip-flop that has been set is reset. That action in turn resets the flip-flop MIP.

Operation of the multiplier sequence control unit 74 (FIG. 7 responds to a change in the output of the comparator 57, i.e., in response to CMP no longer being true, when either $\dot{X}$ or $\dot{Y}$ has changed, in a manner similar to the foregoing. As in the other cases, a separate storage flip-flop is provided to select or control the appropriate one of several predetermined multiplier sequences. However, the output of the flip-flop MIP is inhibited from initiating the selected multiplier sequence by the signal CMP until the square root generator 17 has computed a new value of ballistic velocity (VB), at which time the control logic 59 (FIG. 6) again generates the CMP to allow initiation of the selected multiplier sequence. The sequence selected is like the sequence selected by an increment ΔEXP .

Only one of the three input signals $\Delta\dot{X}$, $\Delta\dot{Y}$ and ΔEXP may initiate a multiplication sequence at any one time. To assure that, those input signals which may set the flip-flop MIP, and which simultaneously set one of three separate storage registers, are gated into input terminals of the flip-flop MIP and the various ones of

the storage registers by AND gates having as additional input control terms the signals CLT and $\overline{\text{MIP}}$. The signal CLT assures that the flip-flop MIP is used only during the computation mode CLT. The other control term $\overline{\text{MIP}}$ then locks out all other input signals to the flip-flop MIP except $\overline{\text{CMP}}$ once one of the three signals $\Delta\dot{X}$, $\Delta\dot{Y}$ and ΔEXP have set. The logic equation for setting the flip-flop MIP and the respective storage registers are then as follows:

$$\text{MIP} = \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\dot{X} + \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\dot{Y} + \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\text{EXP} + \text{CLT} \cdot \overline{\text{MIP}} \cdot \overline{\text{CMP}}$$

$$\Delta\dot{X} \text{ Storage} = \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\dot{X}$$

$$\Delta\dot{Y} \text{ Storage} = \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\dot{Y}$$

$$\Delta\text{EXP Storage} = \text{CLT} \cdot \overline{\text{MIP}} \cdot \Delta\text{EXP}$$

$$\overline{\text{CMP Storage}} = \text{CLT} \cdot \overline{\text{CMP}}$$

As noted hereinbefore, the signal MIP inhibits clock pulses from the integrators 19, 20, 21 and 22, but not from the square root generator (FIG. 6) or the multiplier (FIG. 7).

Before a particular problem can be solved by use of the present invention, the computer must be properly scaled because it can operate with only integrands equal to or less than one. As noted hereinbefore with reference to FIG. 2, the overflow rate of an integrator is proportional to the input rate of increment; the maximum overflow rate obtainable is therefore equal to the input rate. This maximum rate occurs when the integrand is equal to the largest number the integrand register (or counter) can contain. The integrand is then equal to ± 1 , and for an output rate less than the input rate, the integrand is less than one in absolute value. For that reason, any particular integrand value is always less than or almost equal to one, and computer elements must be scaled accordingly.

Before the computer can be scaled, the scaler (scaling factor) must be determined by first calculating all maximum values in proper units. For example, upon selecting a weighted clock Δt equal to 2 milliseconds, a ballistic trajectory problem may be scaled in terms of yards per seconds. Then the maximum value of $\dot{X}$ obtained from the aircraft performance data is used to determine terminal velocity of a weapon for a given height. For example, if maximum horizontal velocity $\dot{X}$ of the aircraft is 1000 ft/sec and height Y is to be 8000 feet, terminal horizontal and vertical velocities can be determined from ballistic tables to be $\dot{X}_{hd} T = 0.8 \dot{X}_{max}$ and $\dot{Y}_T = 0.7 \dot{X}_{max}$. The maximum terminal velocity VB of the weapon can then be found to be

$$VB_T^2 = 0.64 \dot{X}_{max}^2 + 0.49 \dot{X}_{max}^2 = 1.13 \dot{X}_{max}^2$$

$$VB_T = 1.13 \dot{X}_{max} = 3.99 \text{ yds/sec}$$

By rounding out the maximum terminal velocity terminal VB_T at 400 yds/sec, VB_T^2 is then at a maximum of 150,000 (yds/sec)². This allows about 20 percent over and above the actual VB_T^2 .

The maximum constant $K_{hd} B$ is then determined and used to find the maximum horizontal acceleration to be

$$\ddot{X}_{max} = D_B - VB_T \dot{X}_{max} \approx 291 \text{ yds/sec}^2$$

For determining the maximum exponential, the constant $1/b$ is used with maximum constant K_B . Equation (5) may be used for determining the maximum constant K_B by using for the constant K_D the maximum

Mach number of the weapon at the maximum terminal velocity.

Having determined maximum values in the horizontal direction, scaling for the horizontal integrators may be accomplished. In scaling it is necessary to determine the increments by which ground range and altitude are to be accumulated. The following discussion assumes such accumulations are in one yard increments. If n yards per increment prove to be more desirable, where n is any integer, the computer must be rescaled, but the method of scaling remains the same.

Before proceeding with a description of the scaling method, some further observations should be made. If the maximum integrand must be equal to or less than one according to Equation (14), the maximum of all of the following must be equal to or less than one: X , $\dot{X}$, $|1/b|$ and $|K_B e^{-Y/b}|$. It should also be noted that the increment ΔX is one yard, and the input X is expressed in yards per 2 milliseconds. The maximum values of $VBhd$ 1, $\dot{X}$ and $\ddot{X}$ are also expressed in the same units. Having expressed the maximum horizontal velocity in proper units, the number is converted from its decimal form to an octal form as follows:

$$\dot{X}_{max} = (.676)_{10} = (.532)_8$$

Upon converting the octal digits into binary form, it is seen that the maximum horizontal velocity will be the binary number (.101011010) having nine significant digits. Accordingly, the counter for the integrand $\dot{X}$ must be scaled with the least significant bit weighted 2^{-9} and the most significant bit weighted 2^{-1} . Then an overflow from the integrator 22 will be properly weighted 2^{-1+1} , which is equal to 2^0 or one unit (1 yard).

Since the overflow of the integrator 19 accumulated by the counter for the integrand $\dot{X}$ in the integrator 22 must be weighted equal to the least significant bit of the integrand $\dot{X}$, which is 2^{-9} , the most significant bit of the register for the integrand $\dot{X}$ in the integrator 19 must be weighted 2^{-10} . To determine the weight of the least significant bit, it is then necessary to simply look at the value of $\dot{X}_{max}$ in binary form. Since that is a number having nine binary digits, the least significant digit of the register for the integrand $\dot{X}$ is 2^{-18} .

To scale the integrator 28, the maximum value of $\ddot{X}^2$ is translated into binary form and found to have its most significant of seven binary digits weighted 2^{-2} . Recalling Equation (24), and remembering that the maximum value of an integrand must be equal to or less than 1, the value of the integrand $2 \dot{X}_0$ to be stored in the counter of the integrator 28 must be equal to or less than 1. Therefore, $\dot{X}_0$ must be weighted half the original weight, and the weight of the overflow $\Delta \ddot{X}^2$ is equal to $(0.676 - \Delta \dot{X}_0) \Delta \dot{X}_0 = (2^1)$ (weight of $\dot{X}_0$) $= (2^1) (2^{-9}) = 2^{-8}$. The scaled weight of the least significant bit of $\ddot{X}^2$ is 2^{-8} . The weight of the most significant bit of $2\dot{X}$ is then established at 2^0 , and the least significant bit at 2^{-1} , indicating a requirement for a 10-bit counter for the integrand of the integrator 28.

Scaling the integrating functions of the integrators 20, 21 and 29 is similar to scaling the integrators 19, 22 and 28. The difference is only in different maximum values for Y , $\dot{Y}$ and $\ddot{Y}$.

To scale the square root generator 17, the maximum value of $(\dot{X}^2 + \dot{Y}^2)$ is translated into binary form. That indicates the weight of the most significant bit of β^2 is equal to 2^{-1} and the least significant bit of β^2 is equal

to 2^{-8} . At the same time, the most significant bit of the integrand counter 60 must be of equal weight to the most significant bit of $\dot{X}$, which is 2^{-1} . Therefore, the most significant bit of the product 2β is equal to 2^0 , and the overflow is $2^{0+1}\Delta\beta = 2^{-8}$. Hence the weight of $\Delta\beta$ is 2^{-9} . The maximum value of β is $|0.812|_{10} = |2^{-1}|_2$ and the maximum value of 2β is $|1.624|_{10} = |2^0|_2$. Consequently, the register 62 is a 10-bit register, while the counter 60 is of the same number of binary positions as the integrand counter of the integrator 22 for horizontal velocity $\dot{X}$. This assumes maximum horizontal velocity $\dot{X}$ is greater than maximum vertical velocity $\dot{Y}$, a condition which will always be satisfied when V_{ej} is just sufficient to eject the weapon downwardly out of the aircraft's air stream.

To scale the exponential integrators 23 and 16, the maximum value of the constant K_B must be equal to or greater than 2^N , and equal to or less than 2^{N+1} , that is satisfied when $N = -9$. Therefore, the most significant bit of the constant K_B is weighted 2^{-9} . Assuming that when K_B is translated into a binary number it has 9 significant binary digits, the least significant binary digit is weighted 2^{-17} . Therefore, the weight of the overflow of the integrator 23 is 2^{-9} . However, if $1/b = 1/11,000$ yds. $= 9.0909 \times 10^{-5}$, translation of $1/b$ into binary form indicates its most significant bit must be weighted 2^{-14} . That requires each of the remainder register and adder of the integrator 23 is 2^{-9} . However, if $1/b = 1/11,000$ for the integrand $1/b$, i.e., to include 4 bits weighted 2^{-9} to 2^{-13} beyond the most significant bit weighted 2^{-14} .

The length of each accumulator up-counter not included as part of an integrator is determined by the maximum binary value of the number to be accumulated, such as the time of fall counter 12 and the range counter 15. Similarly, the length of accumulator down-counters is determined by the maximum binary value of the number to be counted down, such as the maximum height for the altitude counter 11.

From the foregoing, it is seen that the present invention is a multi-loop function generator for solving the simultaneous non-linear differential Equations (6) and (7). Those equations are the simplest ballistic equations which assume a cool exponential atmosphere. In accordance with a further feature of the present invention, component parts of the same multi-loop function generator are first used to generate all initial conditions for the equations. Those parts are: the vertical velocity integrator 21 (with a constant integrand) to drive the altitude counter 11 from a height & above a standard data plane; the time-of-flight counter 12 to provide timing for the sequence control function of the multiplier 18; the exponential generator (integrators 23 and 16) which receives ΔY increments until the altitude counter 11 reaches zero; the square root generator 17 and the digital multiplier used to compute $\dot{X}_0^2$, $\dot{Y}_0^2$, $\dot{X}_0$ and $\dot{Y}_0 - G$, and to distribute the computed values where needed as initial conditions for solving the Equations (6) and (7). The multiplier is also used to recalculate $\dot{X}$ and $\dot{Y} - G$ whenever any of the factors affecting those values changes during solution of Equations (6) and (7). All integration is interrupted while the multiplier is in operation. In that regard it should be noted that the function of the integrator 21 may be simulated by simply gating clock pulses in place of ΔY pulses to the altitude counter 11 and the integrator 23 during the initial condition computation mode until the counter 11 reaches zero. That would have the advantage of speed-

ing up generation of the exponential just as though a maximum velocity were stored as the integrand in the integrator 21. Other modifications and variations will occur to those skilled in the art without departing from the true spirit and scope of the present invention.

What is claimed is:

1. A system for determining the ballistic range in an exponential atmosphere of a weapon having a known initial velocity at a known altitude above a target said system comprising:

a clock generator for generating clock pulses;

first computing means an exponential factor relating air density at a given altitude to that at a standard pressure plane;

second computing means responsive to said pulses from said clock generator for solving in real time the simultaneous non-linear differential ballistic equations as a point by point trajectory of said weapon from said known altitude to said target, the range and altitude values defining said trajectory being calculated at times determined by said clock pulses, said second computing means further including means for employing re-computed values of said exponential factor as the computed value of altitude of said weapon decreases from said given altitude to the altitude of said target; and means for terminating operation of said second computing means when the most recent value of altitude calculated by said second computing means equals that of said target, whereby said ballistic range is determined by the corresponding calculated range value.

2. A system as defined in claim 1 wherein: said exponential factor is given by $K_{be}^{-Y/b}$, Y being the altitude above said standard pressure plane; and

said first computing means further comprises

a first digital integrator wherein, for each incremental change in altitude ΔY the value $1/b$ is added to the contents of a first remainder register,

a second digital integrator wherein increments $\Delta Y/b$ from said first integrator cause the value $K_{be}^{-Y/b}$ to be added to contents of a second remainder register,

an exponential counter for storing the current value of the function $K_{be}^{-Y/b}$, said counter including means for causing said current value $K_{be}^{-Y/b}$ to be decremented in response to each increment $\Delta K_{be}^{-Y/b}$

3. A system as defined in claim 2 wherein each time said value $K_{be}^{-Y/b}$ is decremented in said counter, said second computing means calculates new values of horizontal and vertical acceleration to be used in the continued point by point solution of said equations.

4. A system for computing range X of a weapon along a ballistic trajectory from a higher first altitude to a lower, second altitude as a function of time from integration of simultaneous non-linear differential equations, said system comprising:

first means for developing a first term including an exponential factor as function of altitude;

second means for developing a second term equal to the square root of the sum of the square of vertical and horizontal velocities;

third means for formulating the product of said first and second terms, and for formulating horizontal and vertical acceleration values from products of

respective horizontal and vertical velocities with said product;

fourth means for digitally integrating said values of horizontal and vertical accelerations by generating increments of change in horizontally and vertical velocity as a function of time and by accumulating said increments of change in velocity to provide new values of horizontal and vertical velocities;

fifth means for causing said third means to formulate a new horizontal acceleration value for each new value of horizontal velocity provided by said fourth means, and to formulate a new vertical acceleration value for each new value of vertical velocity provided by said fourth means; and

sixth means for digitally integrating said new values of horizontal and vertical velocities by generating increments of change in horizontal and vertical position as a function of time and by accumulating said increments of change in position to provide new values of range and altitude position, whereby said range X is given by the value of range corresponding to the altitude value equal to that of the target.

5. A system as defined in claim 4 including means for causing said second means to develop a new value for said second term whenever the value of the second term ceases to equal the value of the square root of the sum of the squares of current values of horizontal and vertical velocities, and for causing said third means to formulate new values of horizontal and vertical acceleration using the new value of said second term.

6. A system as defined in claim 5, including means responsive to said increments of change for causing said first means to develop a new value of said first term, and for causing said third means to formulate new values of horizontal and vertical acceleration using the new value of said first term.

7. A system for determining ballistic range and time of fall of a weapon having a given initial velocity at a given altitude above a target said system comprising:

a clock pulse generator;

first means responsive to said clock pulses for generating by digital integration instantaneous values of vertical velocity, vertical velocity squared and increments of altitude change above said target from instantaneous values of vertical acceleration;

second means responsive to said clock pulses for generating by digital integration instantaneous values of horizontal velocity, horizontal velocity squared and range from instantaneous values of horizontal acceleration;

third means responsive to said first and second means for continually developing by digital servo the square root of the sum of said vertical velocity squared and said horizontal velocity squared;

fourth means for continually developing an instantaneous value of an exponential of the form $K_{be}^{-Y/b}$ in response to said increments of altitude change, where all except Y are constants and Y is the instantaneous value of altitude above said target; and

fifth means for developing new instantaneous values of horizontal and vertical acceleration in accordance with said equations of motion whenever a selected one of said exponential, said vertical velocity, and said horizontal velocity changes value

sixth means for storing a value equal to said altitude above said target, and for decreasing the stored

value of altitude as said first means determines increments of altitude change to provide an instantaneous value of altitude above said target; and a counter for counting the number of pulses generated by said clock pulse generator; whereby said ballistic range and time of fall are provided by the instantaneous values provided by said second means and said counter respectively when said sixth means provides a zero value for altitude above target.

8. A system as defined in claim 5 further comprising a control sequencer having a "calculate initial condition," wherein the initial value of said vertical acceleration and the initial value of said horizontal acceleration are determined by:

operation of said first and second means in advance of solving said equations from initial values of horizontal velocity and vertical velocity set into said first and second means respectively while holding said first and second means inoperative except for generating vertical velocity squared and horizontal velocity squared, and for generating increments of altitude change;

operation of said third means with vertical velocity squared and horizontal velocity squared as inputs from said first and second means;

operation of said fourth means in reverse to provide a decreasing value of said exponential from a value of K_b in response to said increments of altitude change;

operation of said sixth means using a value of initial altitude relative to a standard pressure plane; and operation of said fifth means in the normal manner for solution of said equations of motion when said stored value of altitude above said standard pressure plane is equal to zero, at which time the value of said exponential being generated by said fourth means is the appropriate value at said altitude relative to said standard pressure plane.

9. A system for determining time of fall of a weapon from a known altitude above target comprising: means for generating clock pulses;

computing means responsive to pulses from said clock means for solving in real time simultaneous non-linear differential equations of motion by integration to obtain corresponding values of altitude and range as a function of real time starting from said known altitude employing an exponential term for the value of air density which increases exponentially from an initial value at said known altitude as said weapon falls;

exponential generating means for developing the initial value of said exponential term at said known altitude; and

a counter for counting said clock pulses during the operation of said computing means whereby time of fall is given by said counter when said computing means indicates an altitude equal to that of the target.

10. A system as defined in claim 9 wherein said exponential term is $K_b e^{-Y/b}$, is a known value, at a standard pressure plane b is a known constant selected for the best approximation of air density $\rho = \rho_0 e^{-Y/b}$, Y is instantaneous height with respect to said standard pressure plane, and ρ_0 is air density at said standard pressure plane, and wherein said exponential generating means for developing said initial value of said exponential term comprises a first integrator responsive to increments ΔY of said altitude to develop increments $\Delta Y/b$, a second integrator coupled to said first integrator for developing output increments in response to said increments $\Delta Y/b$ and an integrand stored therein, and means responsive to said output increments to decrease said stored integrand from the value of said constant K_b before the first increment $\Delta Y/b$ to the value $K_b e^{-Y/b}$ when Y is said known altitude by decrementing the stored integrand of said second integrator in response to each of said output increments.

11. A system as defined in claim 10 including means for terminating operation of said exponential generating means when said first integrator has responded to a total number of increments equivalent to said altitude above said standard pressure plane whereby said initial value of said exponential term $K_b e^{-Y/b}$ is provided by said stored integrand of said second integrator.

12. A system as defined in claim 10 wherein said first and second integrators of said exponential generating means are utilized by said computing means to develop said exponential term by developing increments $\Delta Y/b$ in response to increments ΔY , developing output increments from said second integrator in response to said increments $\Delta Y/b$ developed by said first integrator, and incrementing the stored integrand of said second integrator in response to said output increments.

13. A system as defined in claim 12 including means for terminating operation of said computing means when a value of altitude above said target computed by said computing means equals zero, whereby the last value of range computed is said ballistic range.

* * * * *

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,748,440 Dated July 24, 1973

Inventor(s) Thomas F. Alexander

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

IN THE CLAIMS:

Claim 1, column 23, line 9, after "target" insert a comma (,);

Claim 1, column 23, line 12, after "computing means" insert --for calculating--;

Claim 2, column 23, line 39, before "the value" insert a comma (,);

Claim 2, column 23, line 40, correct the spelling of "remainder";

Claim 7, column 24, line 44, correct the spelling of "instantaneous";

Claim 7, column 24, line 65, after "value" insert a semicolon (;);

Claim 8, column 25, line 11, delete "5" and insert --7--;

Claim 9, column 25, line 41, after "above" insert --a--;

Claim 10, column 26, line 11, before "is a known" insert --K_B--;

Signed and sealed this 7th day of May 1974.

(SEAL)
Attest:

EDWARD M. FLETCHER, JR.
Attesting Officer

C. MARSHALL DANN
Commissioner of Patents