A New Method for the Alignment of Inertial Navigation Systems

LAWRENCE W. LYONS III*
Massachusetts Institute of Technology, Cambridge, Mass.

Before an inertial navigation system is used, an initialization procedure is required. The initialization procedure consists of alignment of the platform and the entry of position and velocity. A new method for the alignment of inertial navigation systems is presented in this paper. Alignment of the platform is usually accomplished by a procedure known as gyrocompassing. When gyrocompassing is employed, the inertial system is operated in a damped mode and relatively little base motion is tolerated. The well-known Kalman filter algorithm can also be used in a real time computer to align inertial navigation systems. The method discussed in this paper is in some respects similar to the Kalman filter; however, the definition of a covariance matrix is not required. Like the Kalman filter, the technique can be used to align damped and undamped inertial systems. It can be used to align inertial systems on moving or stationary vehicles. It can also be used to compute gyro drift biases in addition to platform misalignment angles. Computer simulations relating to a number of initial conditions and unmodeled error sources are discussed. It is shown that the algorithm can be used to align a system with considerable base motion.

Nomenclature

F = state differential equation coefficient matrix
g = gravitational constant
G = integral of ϕ matrix
I = identity matrix
l = longitude
l_c = longitude computed by inertial system
R = Earth radius
t = time
u = gyro drift vector
u_x = x gyro drift
u_y = y gyro drift
u_z = z gyro drift
x = state vector
α = platform misalignment angle about x axis
β = platform misalignment angle about y axis
γ = platform misalignment angle about z axis
$\delta\lambda = \lambda_c - \lambda$
$\delta l = l_c - l$
$\delta\omega_n = \omega_{nc} - \omega_n$
$\delta\omega_e = \omega_{ec} - \omega_e$
λ = latitude
λ_c = latitude computed by inertial system
ϕ = state transition matrix
Ω = Earth rotation rate
ω_n = north velocity
ω_{nc} = north velocity computed by inertial system
ω_e = east velocity
ω_{ec} = east velocity computed by inertial system

i = time index
m = row index of ϕ matrix
n = column index of ϕ matrix

I. Introduction

ALIGNMENT of inertial navigation systems was historically considered as a special case of damping. The term gyrocompassing refers to damping for the purpose of azimuth alignment. The damping constants customarily used for alignment are large, and the system is therefore susceptible to base motion. Reference 1 has a chapter devoted to the subject of damping.

An early application of Kalman filtering to the alignment and calibration† of inertial navigation systems is due to Brock.[2] Since the writing of Brock's thesis several papers have been published describing various applications of Kalman filtering to inertial navigation systems (for example, Ref. 3). Kalman filtering can be viewed as an application of state space techniques.

In this paper a new method for the alignment and calibration of inertial navigation systems is discussed. Like the Kalman filter it is a state space technique. The specific application considered is the alignment and calibration of an inertial navigation system on a stationary or slowly moving vehicle (the Navy's Deep Submergence Rescue Vehicle). However, the method applies in principle to fast moving vehicles.

II. The DSRV and the Alignment Problem

The Navy's Deep Submergence Rescue Vehicle (DSRV-1) was launched in early 1970. The decision to develop the DSRV was prompted by the loss of the U.S.S. Thresher in 1963. The DSRV is designed to perform the following two functions. 1) Rescue of personnel and material from a distressed submarine. 2) Search of the ocean bottom. This function includes searching for a distressed submarine, and auxiliary missions such as oceanographic research and contour mapping. The C. S. Draper Laboratory, a division of MIT, was responsible for the navigation, communication and control systems. There is a significant amount of interdependence between the systems and their various subsystems.

It was realized at the beginning of the program that a precise attitude device was needed. For navigation, a good attitude device was needed to resolve inputs from one of the navigation sensors (Doppler sonar). For ship's control, attitude information is especially important when mating the DSRV to the hatch of a distressed submarine.

At the beginning of the program, a decision was therefore made to develop a miniature precision gyrocompass (M.P.G.). Later, it was decided to transform the M.P.G. into an inertial navigation system. Basically, this required the addition of computer software to the original design. It is emphasized that the primary function of the DSRV inertial navigation system is attitude information. For the time being, at least, its value as a dead reckon navigation device is minimal. As

Received December 6, 1971; revision received February 29, 1972. The author acknowledges the criticism of T. B. Green, Assistant Director at the C. S. Draper Laboratory, who read the manuscript.
Index category: Navigation, Control, and Guidance Theory.
* Research Assistant, Mathematics Department. Member AIAA.

† Alignment is defined to be the procedure for correct orientation of the platform. Calibration is defined to be the compensation for gyro drifts.

Table 1 Characteristics of DSRV navigation sensor suit KALFIL algorithm

Case	Sensors	Rms error in x and y position	Rms error in x and y velocity
1.	a) Depth sonar and/or depth gage b) Doppler sonar	5 ft	0.1 fps
2.	a) Depth sonar and/or depth gage b) Acoustic transponders	15 ft	1 fps
3.	a) Depth sonar and/or depth gage b) Doppler sonar c) Acoustic transponders	5 ft	0.1 fps

will be seen, the operational constraints of the DSRV are such that alignment of the inertial navigation system is very difficult. Also, the velocity of the vehicle is usually only about 5 fps.

The alignment algorithm for inertial navigation systems discussed in this paper is a general purpose algorithm. Thus it is applicable to the DSRV, where the conditions for alignment are quite unfavorable; and it is also applicable when the conditions for alignment are ideal (for example, an inertial navigation system on a commercial airplane at an airport). This illustrates the principle that when one has solved the general problem, one has also solved the particular problem.

Alignment of an inertial navigation system requires an external indication of true position or velocity. A special case is when an inertial system is stationary on the Earth's surface; in this case the velocity is known to be zero. Alignment of the DSRV inertial system is accomplished on the surface of the ocean. In an actual search or rescue mission the vehicle would be transported to the site by a mother submarine. The vehicle would then be detached from the submarine, the power supply turned on and the vehicle made ready for diving. A short period of time is available to ready the vehicle for diving; in particular, the time alloted to align the inertial system is about 15 min. The inertial system is located off the center of buoyancy of the DSRV; therefore, wind, ocean current and wave motion may result in considerable base motion.‡ Thus the conditions for alignment are unfavorable.

When the DSRV is moving near the ocean bottom, position and velocity will usually be available from a navigation sensor suit. This sensor suit consists of acoustic transponders, Doppler sonar, altitude and depth sonar and two pressure depth gages. Information from these sensors is processed by a Kalman filter algorithm which, for the purpose of discussion, we shall name KALFIL. Position and velocity information from KALFIL can be used to further align the inertial system when the DSRV is moving. The quality of position and velocity information from KALFIL is discussed briefly in Table 1, since this determines how well the inertial system can be further aligned. A geographic coordinate system is assumed: positive x axis to the north, positive y axis to the east and positive z axis down.

III. State Space Equations

The DSRV inertial navigation system is mechanized as follows: three gimbals, two accelerometers, three single degree of freedom gyroscopes. It is a local level mechanization. The equation describing the system error propagation is, to a good approximation for a slowly moving vehicle

$$\dot{x} = Fx + u \tag{1}$$

where

$$x = (\delta\omega_n, \delta\omega_e, \delta\lambda, \delta l, \alpha, \beta, \gamma), \qquad u = (0,0,0,0,u_x,u_y,u_z)$$

$$F = \begin{bmatrix} 0 & -2\Omega\sin\lambda & 0 & 0 & 0 & g/R & 0 \\ 2\Omega\sin\lambda & 0 & 0 & 0 & -g/R & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sec\lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\Omega\sin\lambda & 0 & 0 & -\Omega\sin\lambda & 0 \\ -1 & 0 & 0 & 0 & \Omega\sin\lambda & 0 & \Omega\cos\lambda \\ 0 & -\tan\lambda & -\Omega\cos\lambda & 0 & 0 & -\Omega\cos\lambda & 0 \end{bmatrix}$$

The preceding equation applies when the system is in the undamped (free inertial) mode. See Ref. 1 for a derivation of this equation. Equation (1) is a linear, vector differential equation. We now consider the state space equations for dynamical systems described by such an equation. Consider the equation

$$\dot{x} = Fx + u(t) \tag{2}$$

The solution to Eq. (2) is

$$x(t) = \phi(t)x(0) + \int_0^t \phi(t-\tau)u(\tau)d\tau \tag{3}$$

where $\phi(t)$ is the state transition matrix and is the solution of the matrix differential equation

$$d\phi(t)/dt = F\phi(t), \qquad \phi(0) = I \tag{4}$$

The state transition matrix is the matrix exponential of Ft

$$\phi(t) = e^{Ft} \tag{5}$$

It follows that

$$\phi^{-1}(t) = \phi(-t)$$

and

$$\phi(nt) = \phi^n(t)$$

Also,

$$\phi(t+s) = \phi(t)\phi(s)$$

If each element of u is constant (does not vary with time), Eq. (3) can be simplified. In this case

$$x(t) = \phi(t)x(0) + G(t)u \tag{6}$$

where the G matrix satisfies the differential equation

$$dG(t)/dt = \phi(t), \qquad G(0) = 0 \tag{7}$$

IV. Solution of the Alignment Problem

An inertial navigation system requires a real time computer (analog or digital). The computer integrates accelerometer outputs to obtain position and velocity and also computes gyro torquing signals. Therefore, position and velocity as computed by the inertial system is continuously available. We will assume that readings of position (latitude and longitude) or velocity are taken at n times $t_1, t_2, \ldots, t_n$. We will further assume that an external indication of true position or velocity is available at the same times.

From Eq. (6) one can express $\delta\omega_n(t)$ as a function of time and the initial value of the state vector. Thus

$$\begin{aligned} \delta\omega_n(t_i) = {} & \phi_{11}(t_i)\delta\omega_n(0) + \phi_{12}(t_i)\delta\omega_e(0) + \phi_{13}(t_i)\delta\lambda(0) \\ & + \phi_{15}(t_i)\alpha(0) + \phi_{16}(t_i)\beta(0) + \phi_{17}(t_i)\gamma(0) \\ & + G_{15}(t_i)u_x + G_{16}(t_i)u_y + G_{17}(t_i)u_z \end{aligned} \tag{8}$$

There is no contribution from an initial longitude error, i.e.

‡ Base motion is defined to be random and uniform motion of the inertial navigation system case.

$\phi_{14}(t)=0$ for all t. Similar equations apply for $\delta\omega_e(t)$, $\delta\lambda(t)$ and $\delta l(t)$. Thus

$$\begin{aligned}\delta\omega_e(t_i) &= \phi_{21}(t_i)\delta\omega_n(0)+\phi_{22}(t_i)\delta\omega_e(0)+\phi_{23}(t_i)\delta\lambda(0)\\ &+\phi_{25}(t_i)\alpha(0)+\phi_{26}(t_i)\beta(0)+\phi_{27}(t_i)\gamma(0)\\ &+G_{25}(t_i)u_x+G_{26}(t_i)u_y+G_{27}(t_i)u_z \end{aligned}\tag{9}$$

$$\begin{aligned}\delta\lambda(t_i) &= \phi_{31}(t_i)\delta\omega_n(0)+\phi_{32}(t_i)\delta\omega_e(0)+\phi_{33}(t_i)\delta\lambda(0)\\ &+\phi_{35}(t_i)\alpha(0)+\phi_{36}(t_i)\beta(0)+\phi_{37}(t_i)\gamma(0)\\ &+G_{35}(t_i)u_x+G_{36}(t_i)u_y+G_{37}(t_i)u_z \end{aligned}\tag{10}$$

$$\begin{aligned}\delta l(t_i) &= \phi_{41}(t_i)\delta\omega_n(0)+\phi_{42}(t_i)\delta\omega_e(0)+\phi_{43}(t_i)\delta\lambda(0)\\ &+\phi_{45}(t_i)\alpha(0)+\phi_{46}(t_i)\beta(0)+\phi_{47}(t_i)\gamma(0)\\ &+G_{45}(t_i)u_x+G_{46}(t_i)u_y+G_{47}(t_i)u_z \end{aligned}\tag{11}$$

The first stage of alignment of inertial navigation systems is accomplished by comparing the gravity and Earth rate vectors. This takes about one minute of time. It is assumed that this first stage of alignment has been effected for application of the algorithm described herein. The azimuth is thus determined to within a few degrees, and the verticals to within a few arc minutes.

Consider first Eqs. (8) and (9). In Eq. (8), note that $\delta\omega_n(t_i)$ is known for all t_i. This is true because we have assumed that both the inertial system north velocities and the true north velocities are known for all t_i. (Recall the definition $\delta\omega_n = \omega_{nc}-\omega_n$). To be more precise, $\delta\omega_n(t_i)$ is known within certain limits; the limits are determined by such factors as quantization noise. Also note that $\phi_{mn}(t_i)$ can be computed for all i,m,n. The same argument applies to Eq. (9). Therefore the quantities $\alpha(0)$, $\beta(0)$, $\gamma(0)$, $\delta\lambda(0)$, u_x, u_y, and u_z are the unknowns and the other quantities are known. Thus if a sufficient number of readings of $\delta\omega_n(t)$ and $\delta\omega_e(t)$ are taken we can solve for the unknowns in terms of the knowns. In practice, singularities will arise if one attempts to solve for $\delta\lambda(0)$ and u_y; this is discussed below. However, one can solve for the initial misalignment angles $\alpha(0)$, $\beta(0)$, and $\gamma(0)$ and the gyro drift biases u_x and u_z. Once the initial state is computed the final state is computed from the formula

$$x(t_f)=\phi(t_f)x(0)+G(t_f)u \tag{12}$$

The time t_f is the system reset time; $t_f \geq t_n$. The time t_n is the last time at which readings of $\delta\omega_n$ and $\delta\omega_e$ are taken.

For a local level inertial navigation system, it is known that a limiting factor in azimuth alignment is the y gyro drift. Also, a latitude error is a limiting factor in the determination of the x gyro drift. The singularities involving $\delta\lambda(0)$ and u_y are thought to be reflections of these facts. The singularity problem is solved by elimination of the terms involving $\delta\lambda$ and u_y from Eq. (8) [$\phi_{13}(t_i)\delta\lambda(0)$ and $G_{16}(t_i)u_y$]. The corresponding terms are also eliminated from Eqs. (9,10 and 11). The elimination of these terms has a slight effect on alignment accuracy; this is shown by the computer simulation results discussed later. The F matrix in Eq. (1) is *not* modified to account for the deletion of the $\delta\lambda$ and u_y terms. Thus, the ϕ matrix in Eq. (6) is not modified.

We first consider a deterministic example. From Eqs. (8) and (9) one has

$$\begin{aligned}\delta\omega_n(t_1) &= \phi_{11}(t_1)\delta\omega_n(0)+\phi_{12}(t_1)\delta\omega_e(0)\\ &+\phi_{15}(t_1)\alpha(0)+\phi_{16}(t_1)\beta(0)+\phi_{17}(t_1)\gamma(0)\\ \delta\omega_n(t_2) &= \phi_{11}(t_2)\delta\omega_n(0)+\phi_{12}(t_2)\delta\omega_e(0)\\ &+\phi_{15}(t_2)\alpha(0)+\phi_{16}(t_2)\beta(0)+\phi_{17}(t_2)\gamma(0)\\ \delta\omega_e(t_2) &= \phi_{21}(t_2)\delta\omega_n(0)+\phi_{22}(t_2)\delta\omega_e(0)\\ &+\phi_{25}(t_2)\alpha(0)+\phi_{26}(t_2)\beta(0)+\phi_{27}(t_2)\gamma(0) \end{aligned}\tag{13}$$

The contribution from the gyro drifts has been neglected, and only the first order contribution from the misalignment angles is included. Since $\delta\omega_n(0)$, $\delta\omega_n(t_1)$, $\delta\omega_n(t_2)$, and $\delta\omega_e(0)$ and $\delta\omega_e(t_2)$ are known we can solve for $\alpha(0)$, $\beta(0)$, and $\gamma(0)$. Then we find $\alpha(t_2)$, $\beta(t_2)$ and $\gamma(t_2)$ from the formula

$$x(t_2)=\phi(t_2)x(0) \tag{14}$$

In formula (14) $\delta\lambda(0)$ and $\delta l(0)$ are set equal to zero in the initial state $x(0)$. The system is then reset: the roll, pitch and azimuth gimbals of the platform are torqued by the amounts $-\alpha(t_2)$, $-\beta(t_2)$ and $-\gamma(t_2)$ respectively. One can also subtract the computed values of $\delta\omega_n(t_2)$, $\delta\omega_e(t_2)$, $\delta\lambda(t_2)$ and $\delta l(t_2)$ from the values of ω_n, ω_e, λ, and l in the inertial system computer.

The general case is when readings of $\delta\omega_n$ and $\delta\omega_e$ are taken at n times $t_1, t_2, \ldots, t_n$; α, β, γ, u_x, and u_z are to be computed and the system is to be reset at time t_f where $t_f \geq t_n$. In this case, instead of a deterministic equation set such as Eq. (13), one has an overdetermined set of linear equations of the form

$$v = Hw \tag{15}$$

A least squares solution of this equation is

$$w=(H^TH)^{-1}H^Tv \tag{16}$$

The final state $x(t_f)$ is computed by means of Eq. (12) and the system is then reset. The computation of u_x and u_z is optional; if these are computed u_y is set equal to zero in Eq. (12).

We have so far considered the case where an inertial navigation system is to be aligned and calibrated by means of velocity readings only. That is, the discussion has been confined to Eqs. (8) and (9).

The extension of the discussion to Eqs. (10) and (11) is obvious. That is, an inertial navigation system can be aligned and calibrated by means of position readings. In the next section computer simulations of the alignment algorithm outlined previously are discussed. In particular, we consider the case where there is considerable base motion. Here it is advantageous to use position readings.

V. Computer Simulations

The computer simulations discussed below are divided into four categories.

1) Alignment from velocity readings [Eqs. (8 and 9)]. a) Determination of α, β, γ; b) determination of α, β, γ, and u_x.

2) Alignment from position readings [Eqs. (10 and 11)] in which $\delta\omega_n(0)$ and $\delta\omega_e(0)$ are considered as known quantities. a) Determination of α, β, γ; b) determination of α, β, γ and u_x.

3) Alignment from position readings [Eqs. (10 and 11)] in which $\delta\omega_n(0)$ and $\delta\omega_e(0)$ are considered as unknown quantities. a) Determination of α, β, γ

4) Two special simulations which correspond to alignment of an inertial navigator at an airport. a) α, β, γ, and u_x are determined.

The distinction between categories 2 and 3 is subtle, but is based upon the fact that if one knows $\delta\lambda(t)$ and $\delta l(t)$ within certain limits one also knows $\delta\omega_n(0)$ and $\delta\omega_e(0)$ within certain limits. Thus one has the option of treating the initial velocity errors as knowns or unknowns in Eq. (10 and 11). It is shown, however, that it is better to consider $\delta\omega_n(0)$ and $\delta\omega_e(0)$ as unknown quantities when there is significant base motion.

The conditions common to simulations in each category are listed as follows. 1) For all simulations in each category the initial values of α, β and γ were 1 arc minute, 1 arc minute and 3°, respectively. Also, the latitude for all simulations was 45°. 2) For all simulations in category 1, 15 readings of $\delta\omega_n$ and $\delta\omega_e$ were assumed to be taken one minute apart. 3) For all simulations in categories 2 and 3, 15 readings of $\delta\lambda$ and δl were assumed to be taken one minute apart. 4) For the two category 4 simulations, readings of $\delta\omega_n$ and $\delta\omega_e$ were taken at $t=4$ minutes and $t=8$ minutes.

Conditions which vary with the individual simulation are as follows. 1) Initial values of $\delta\omega_n$, $\delta\omega_e$, $\delta\lambda$, and δl. 2) rms

Table 2 Computer simulation results

			Errors in			
Case	Category	Notes*	α (arc sec)	β (arc sec)	γ (arc min)	$u_{x(\text{meru})}$
1	1a	a,d,h,l,o	0.	0.	0.	
2	1a	a,d,i,l,o	−1.5	−2.1	−2.4	
3	1a	a,d,i,m,o	10.1	−4.1	−8.9	
4	1a	b,e,i,m,p	3.5	−2.1	−10.0	
5	1b	a,d,i,m,o	−2.1	−4.1	−8.9	−0.07
6	2a	c,d,j,l,o	0.	0.	0.	
7	2b	b,d,j,n,o	0.	0.	0.	0.
8	2a	c,g,k,m,o,q	1.7	−43.3	−37.5	
9	3a	c,g,k,m,o	1.2	6.2	−2.1	
10	3a	c,g,k,m,p	0.9	6.2	−4.5	
11	3a	c,d,j,l,o	0.	0.	0.	
12	4	c,r,s,m	0.2	0.	−6.5	0.
13	4	c,d,s,m	0.	0.	−6.5	0.

* a) $\delta\omega_n(0) = \delta\omega_e(0) = 0$; b) $\delta\omega_n(0) = 21$ fps, $\delta\omega_e(0) = -21$ fps; c) $\delta\omega_n(0) = 2.1$ fps, $\delta\omega_e(0) = -2.1$ fps; d) $\delta\lambda(0) = \delta l(0) = 0$; e) $\delta\lambda(0) = 2$ miles, $\delta l(0) =$ arbitrary; f) $\delta\lambda(0) = 2$ miles, $\delta l(0) = 2$ miles; g) $\delta\lambda(0) = -2$ miles, $\delta l(0) = -2$ miles; h) rms noise in $\delta\omega_n$ and $\delta\omega_e = 0$; i) rms noise in $\delta\omega_n$ and $\delta\omega_e = 0.1$ fps; j) rms noise in $\delta\lambda$ and $\delta l = 0$; k) rms noise in $\delta\lambda$ and $\delta l = 21$ ft; l) $u_x, u_y, u_z = 0$; m) $u_x, u_y, u_z = 1.4$ meru; n) $u_x = 1.4$ meru, $u_y = 0$, $u_z = 0$; o) north and east currents = 0; p) north and east currents = 0.73 fps; q) error of 1.1 fps in $\delta\omega_n(0)$ and $\delta\omega_e(0)$; r) $\delta\lambda(0) = 0.5$ miles, $\delta l(0) = -0.5$ miles; s) rms noise in $\delta\omega_n$ and $\delta\omega_e = 0.004$ fps.

noise in $\delta\omega_n$, $\delta\omega_e$, $\delta\lambda$, and δl. The noise was computed by means of a normally distributed random number generator. 3) Values of gyro drifts u_x, u_y, u_z. Note, unless otherwise noted, gyro drifts are constant for all simulations discussed. 4) Ocean current.

The conditions applicable to each simulation are referenced by footnotes to Table 2. The table contains the errors in the computed values of α, β, γ, and u_x at $t = 15$ min (for category 4, $t = 8$ min). These are the errors which apply at the time the inertial navigation system is reset. The errors at $t = 0$ have been found to be of the same magnitude.

The results for cases 1, 6, 7, and 11 demonstrate the analytic nature of the algorithm. That is, when there are no unmodeled errors exact results are obtained. The alignment errors in case 2 are due to the noise in the reference velocities. Case 3 illustrates the larger errors which result when unmodeled gyro drift biases are present. In particular, the difference between the azimuth errors in cases 2 and 3 is due to the y gyro drift present in case 3. Case 4 demonstrates that the effect from small initial latitude errors is slight. Case 5 is identical to case 3 except that u_x is modeled. When u_x is computed, the error in α is reduced. Case 8 illustrates the large errors induced by large errors in the initial velocities. This was also found to be true when the velocity version of the algorithm is used. That is, when the rms noise in $\delta\omega_n(t)$ and $\delta\omega_e(t)$ is 1.1 fps, errors comparable to those in case 8 will result. Cases 9 and 10 demonstrate the errors which result when position alone is used as a reference. Note that the noise in $\delta\lambda$ and δl arises from two sources: quantization errors and errors in the external position reference. Quantization errors can be reduced by extended precision in the latitude and longitude registers. With regard to precision, note that only the low order parts of the latitude and longitude registers are necessary for inputs to the algorithm. The cases (4 and 10) in which an ocean current was simulated demonstrate the fundamental difference between the effects of random and uniform motion of the base. Thus an ocean current of 0.73 fps induces a corresponding error in the estimate of azimuth; estimation of the vertical errors is essentially unaffected. Cases 12 and 13 are interesting because they illustrate the excellent alignment which can be obtained when the only unmodeled errors are the y and z gyro drifts. These simulations were deterministic; readings of $\delta\omega_n$ and $\delta\omega_e$ were taken at $t = 0$, $t = 4$, and $t = 8$.

When the position version of the algorithm is used [Eqs. (10) and (11)] the $\delta\lambda$ and δl readings are relative. Thus the actual readings are $\delta\lambda - \delta\lambda(0)$ and $\delta l - \delta l(0)$, not $\delta\lambda$ and δl. This was taken into account in the simulations.

A few simulations were run in which there were small random components in the gyro drifts. There were no essential differences from the results set forth in the table. When the vehicle is moving, an external position or velocity reference is necessary. The external reference will have its own vertical and azimuth errors. In the case of the DSRV, the Doppler sonar is resolved by the inertial navigation system. Thus the azimuth error of the Doppler is the azimuth error of the inertial navigation system. For a vehicle which moves slowly in a straight line, small biases in the reference velocities result.

The simulations demonstrate that for a system with considerable random base motion, the position alignment algorithm should be used. Specifically, the initial velocity errors are treated as unknowns. The essential steps in the solution of the DSRV's alignment problem, then, were as follows: 1) Formulation of the alignment algorithm, primarily by Eqs. (8–11); 2) The recognition that position readings will give better results than velocity readings will; 3) The knowledge that an inertial navigation system can be aligned in the free inertial mode. This was due to the author's experience in Kalman filtering. Position is the integral of velocity, and integration is a smoothing procedure. Therefore, other factors being equal, $\delta\lambda$ and δl will have less of a noise content than will $\delta\omega_n$ and $\delta\omega_e$. This appears to be the reason why the use of position is better.

VI. Implementation in Real Time

The precise manner in which an algorithm is implemented in a real time computer depends on several factors. These include instruction repertoire, input/output capabilities and available software packages (e.g., library subroutines). If fixed point (scaled) arithmetic is to be used for an algorithm such as that described in this paper, a considerable amount of scaling analysis is usually required. If floating point hardware or software is used, and if a flexible library subroutine package exists, the implementation is easier.

We note that the initial state $x(0)$ is computed in order to determine the final state $x(t_f)$. This is useful for software testing. Thus to test the software package one may first align the inertial system by gyrocompassing. After this is done, the roll, pitch and azimuth gimbals are torqued by the amounts $\alpha(0)$, $\beta(0)$, and $\gamma(0)$. The system is then put in the free inertial mode, and the algorithm then applied. If the software package is correct, one should be able to determine $\alpha(0)$, $\beta(0)$, and $\gamma(0)$. Depending on the $I/0$ capabilities, it may be desirable to display the reset quantities $\alpha(t_f)$, $\beta(t_f)$, and $\gamma(t_f)$.

The computation of the state transition matrix elements can be accomplished as follows. The matrix differential equation

$$\dot{\phi} = F\phi, \qquad \phi(0) = I$$

is solved to find $\phi(t_1)$ for a fixed t_1. This can be done by one of the Runge-Kutta methods (see Ref. 4). A property of the state transition matrix will aid the computational procedure if the points $t_1, t_2, \ldots, t_n$ are equally spaced. This property is

$$\phi(nt) = \phi^n(t)$$

Thus one computes $\phi(2t)$ by computing the square of $\phi(t)$ etc.

When one has computed all of the pertinent elements of the state transition matrices, the next step is to compute the quantity

$$(H^T H)^{-1} H^T$$

This will, in general, require matrix multiplication and matrix inverse subroutines. If the $I/0$ capabilities include tape input, overlay procedures are applicable. Thus one first computes the elements of the state transition matrices. One then computes $(H^T H)^{-1} H^T$. The next step is the computation of the initial state vector. The final step is the computation of the reset state vector.

VII. Additional Considerations

During a 15 min interval of time, the response of an inertial navigation system to a z gyro drift on the order of 1 meru is slight. A simulation was run in which the sensitivity of the z gyro drift computation was demonstrated. The only error source was quantization noise. The misalignment angles α, β, γ and the x gyro drift were well-determined but the z gyro drift was not. Thus it seems that a time interval of at least one hour is needed to compute the z gyro drift. For this reason, the z gyro drift was not computed in the simulations discussed in Table 2. When the z gyro drift is not computed, one sets $u_z = 0$ in the reset Eq. (12).

The reader familiar with Kalman filter techniques is probably interested in a comparison with the techniques described in this paper. One advantage of the algorithm is that it is unnecessary to model the environment in detail. In particular, one includes only the state variables to be computed. As was mentioned earlier, wind, ocean current and wave motion will determine the DSRV inertial system's base motion during a surface alignment. At present, very little is known about the quantitative nature of this effect. With a Kalman filter, it is necessary to define a covariance matrix; in particular its initial value. Also, it is necessary to define the variances associated with sensor measurements. If one expects large vertical and azimuth errors, the initial value of the covariance matrix will reflect this. However, this extends the time required to converge to the vertical and azimuth errors. Thus the initial value of the covariance matrix and the measurement variances affect the filter convergence characteristics.

For a small number of velocity (or position) measurements the algorithm will require less computer memory and time than an equivalent Kalman filter. When velocity instead of position measurements are used, the computer memory and time requirements are even less (the dimension of the matrix to be inverted is 3×3 instead of 5×5). A thorough discussion of computer memory and time requirements would be quite complicated. The memory and time requirements for both the Kalman filter and the algorithm discussed in this paper depend on the specific computer and computer program employed. To illustrate this point, we will consider one of the computational steps: the computation of the state transition matrix. There are at least four methods applicable here; 1) Runge Kutta, 2) Gill, 3) Matrix exponential, 4) closed form expressions for the state transition matrix elements. The author has recently derived these formulas. To summarize, the computer benefits obtained from implementing the algorithm depend on the specific hardware and software used.

The algorithm is theoretically limited to constant gyro drifts when the x and z gyro drifts are to be computed. The Kalman filter, unlike this algorithm, has the ability to track time varying gyro drifts. Thus there are two questions relating to gyro drifts: 1) the effect of time varying gyro drifts on alignment accuracy, and 2) the computed values of the x and z gyro drifts when the actual x, y, and z gyro drifts vary.

A few computer simulations were run in order to answer these questions. With regard to the first question, alignment accuracies are determined primarily by the average values of the gyro drifts. With regard to the second question, the average value of the x gyro drift is computed. (The z gyro drift was not computed; the reason for this was discussed previously.) We note that over a 15 min time interval the variation in gyro drifts is usually small. Also, it is interesting to note that the standard three fix reset method is theoretically limited to constant gyro drifts.

The specific inertial navigation system considered in this paper is a local level mechanization. However, the algorithm applies to other mechanizations as well. To apply the algorithm to other mechanizations, it is necessary to define the canonical matrix (F matrix) for the system error equations. See Ref. 5 for the canonical matrices applicable to various mechanizations.

An intriguing question is how fast an inertial navigation system can be aligned. It should be noted that there was no assumption made with respect to the time intervals between velocity (or position) readings. The author is now conducting a formal error analysis of the alignment algorithm. The effects due to nonlinearities and unmodeled error sources will be studied. These effects may have an important bearing on the question of alignment time.

It may be asked whether the algorithm can be extended to include the computation of parameters such as gyro scale factors and accelerometer misalignment angles. (This would be practical only on a test stand.) One limiting factor would be the variation in time of the gyro drifts. So formally, one could ask: what amount of variation in the gyro drifts can be tolerated to extend the algorithm.

When an inertial navigation system is to be aligned on a fast moving vehicle (for example, an aircraft in flight) the analysis is complicated by two factors. First, the F matrix in Eq. (1) is modified to include the effects of the vehicle's velocity. Second, if the external position or velocity reference is resolved by the inertial navigation system, large bias errors can result. Thus

$$v_x' \approx v_x \cos\gamma + v_y \sin\gamma; \qquad v_y' \approx -v_x \sin\gamma + v_y \cos\gamma$$

where v_x' and v_y' are the indicated x and y velocities and v_x and v_y are the true x and y velocities. The bias error in the x velocity is therefore equal to the product of the y velocity and the azimuth error.

When an inertial navigation system is operated in a damped mode, the canonical matrix (F matrix) for the system error equations is modified. Two velocity elements are added to the state vector to account for vertical damping. Extension of the alignment algorithm to damped inertial navigation systems is therefore straightforward.

Since the writing of the original manuscript, the author has derived a system of nonlinear differential equations describing the platform dynamics, valid for large azimuth misalignment angles. These equations are a generalization of equation set (1), and in fact reduce to set (1) when small angle approximations are made for $\cos\gamma$ and $\sin\gamma$. Briefly, the nonlinearities arise as follows. In the derivation of equation set (1), one neglects products of angles. This is invalid for large angles. The effects due to nonlinearities are noticeable only when the azimuth error is greater than one degree.

The effects due to nonlinearities can be minimized by applying the algorithm on an iterative basis. Here one computes the misalignment angles (and the x gyro drift), resets the system and repeats the procedure. The first reset will remove the large azimuth error. The equation set (1) will then be valid for the second computation of the misalignment angles. We note that the ability to rapidly reset an inertial navigation system depends on high-torque windings.

References

[1] Broxmeyer, C., *Inertial Navigation Systems*, McGraw-Hill, New York, 1964.

[2] Brock, L. D., "Application of Statistical Estimation to Navigation Systems," T-414, Ph.D. thesis, June 1965, MIT Instrumentation Lab., Cambridge, Mass.

[3] Leondes, C. T., *Theory and Applications of Kalman Filtering*, NATO AGARDograph 139, London, 1970.

[4] Hildebrand, F. B., *Introduction to Numerical Analysis*, McGraw-Hill, New York, 1956, pp. 233–239.

[5] Britting, K. R., "Unified Error Analysis of Terrestrial Inertial Navigation Systems," AIAA Paper 71-901, Hempstead, N.Y., 1971.