INS Alignment Using Onboard Computer Software for Missile Trajectory

Dr/ Ibrahim Ismail Arafa, Dr/ Ahmed Mohsen, Eng/ A.Elborae
Guidance Department, Military Technical Collage (M.T.C.), Cairo, Egypt

Abstract

In this paper, the onboard computer software program was developed to predict the full trajectory of the missile in order to improve the accuracy at impact. For system in which time is not critical, the alignment process can be performed prior to launch by utilizing navigation data from the launch platform and calibrating the missile’s INS to this reference. However, for systems that require rapid reaction time, significant pre-launch delays for alignment are not tolerable. In addition, due to low quality of the INS and its life cycle is short; hence every aborted mission will bring the INS closer to its mean time between failures (MTBF). These problems can be minimized by in-flight alignment (IFA). While IFA may seem to be less accurate and more complicated than alignment in the rest, it turns out that the ability of the carrier to maneuver during the IFA phase is a blessing in disguise since it enables the excitation of latent modes and thus, enhances their observability of the whole INS. The effects that maneuvers have on the estimability of the INS states during IFA were investigated in the past. But until now, there are many questions such as: what is the degree of observability? What limits the estimability of the system? What are the minimal additional measurements needed to turn the system into a completely observable one? And what are the useful numbers of maneuvers in IFA? , were not completely addressed. These questions are answered through this paper. We introduce the error model, which appropriately describes the INS behavior during IFA. We then substantiate its adequacy for consideration as a piece-wise constant system (PWCS) and then the simpler stripped observability matrix (SOM) can be used in the system observability analysis.

Introduction

The INS navigation loop provides continuous and reliable navigation solutions to the guidance and flight control loop for autonomous flight. With additional air data and engine thrust data, the guidance loop computes the guidance demands to follow way-point scenarios. The flight control loop generates actuator signals for the control surfaces and thrust vector. The whole Guidance, Navigation and Control (GNC) algorithm was implemented within an embedded flight control
computer. The real-time flight test results show that the vehicle can perform the autonomous flight reliably with high accuracy [1].

The Onboard computer software program was developed to predict the full trajectory of the missile in order to improve the accuracy at impact. The primary purposes of missile launch surveillance are: 1) to provide a timely report of each occurrence of a missile launch, 2) to estimate launch/trajectory parameters, and 3) to estimate present and future missile trajectories as a function of time during flight. After the missile position and velocity vectors are obtained, the prediction starts with the time integration of equations of motion using a six parameter-state vector, which is composed of the position and velocity vectors, [2].

The integration procedure continues until the criterion of the missile range control is satisfied through onboard computer software. The guidance loop generates the guidance commands from the vehicle states and the desired waypoint information. It computes required vehicle speed with respect to the air, height and bank angle. Then the flight control loop (or autopilot) generates actuator control signals to make the vehicle follow the guidance demands as well as to stabilize the vehicle. The control outputs are fed to the control surfaces, or aileron, elevator and rudder, and thrust vector [3, 4].

1) Program objective (mission)

The basic program objective is to fly a typical mission profile as shown in fig (1) with an on target accuracy of $\pm 200m$. The flight path profile in the active phase approximately 59 sec. The reminder of 3 minute flight is a ballistic trajectory. The command of separation will arrive after the shut-off time of the engine within 5 sec. The payload objects are deployed in reentry phase over a distance about 25 km from the ground. To meet target accuracy, the navigation system must provide accurate position and velocity input to the overall Guidance Navigation and Control System (GNCS) for the active phase of flight [5]. At the end of active phase, the program has no other propellant capability so the GNCS has no more control over the delta velocity of the payload objects.
The navigation solution must therefore be highly accurate for the first 59 sec of flight (active phase) and thereafter, it provides useful information about the shape of the trajectory actually flown.

2) On-board Software

The On-Board Software (OBS) is a complex system with many functions. It performs the following functions as shown in fig (2): interface with ground support equipment, the majority of system test and checkout, ground processing, launch and all of the navigation, guidance, and control and sequencing functions required for flight. The OBS development was comprised of 3 major components: 1) the operating system, IMU digital signal processing code and ground alignment software and 2) ground checkout code developed uniquely for the program system and 3) the flight guidance, navigation, control and sequencing code. The flight navigation routines are dependent on inputs from the IMU sensors (gyros, accelerometers) [6]. The compensated sensor outputs, i.e. incremental angles and velocities, are accumulated and made available at 256 [Hz].

Additional compensations are also implemented in the flight inertial navigation algorithm to compensate for the estimated instrument parameters via the System Error Estimator (SEE) updates [6,7]. After the final implementation integrates the guidance kit with the flight navigation solution via the Kalman filter: There are two modes of operation for the system: 1) ground test which assure the navigation system is healthy before launch and 2) navigation in flight.

In the ground test we compute the system azimuth, level orientation and auxiliary sensor compensation through the ground alignment time. This creates the initial attitude reference and gyro and accelerometer biases and scale factors which are used at the time of (Go Inertial) for the flight navigation system [7, 8].
3) Protocol of Mathematical tasks

Here we describe the block diagram of an operating program, functional modules and interrupt handlers as part of the Navigation software. The Navigation mathematical software (NMSW) performs the following functions:

- Support for INS initial alignment
- Solution of navigation problems: inertial velocity and coordinate reckoning.
- Interaction with navigation equipment and external users of navigation data.
- Support for inertial sensors calibration.

The interaction with users and external systems is provided using RS-232 interface. The NMSW is intended for operation as part of INS in the central processor.

a) Navigation Mathematical Software

Transformation matrix between Wander azimuth launch coordinate system and body-fixed system:

\[
M_{bt} = \begin{bmatrix}
\cos \vartheta \cos \psi & \sin \vartheta \cos \psi & \sin \psi \\
-\cos \vartheta \sin \gamma \sin \psi - \sin \vartheta \cos \gamma & -\sin \vartheta \sin \gamma \sin \psi + \cos \vartheta \cos \gamma & \sin \gamma \cos \psi \\
-\cos \vartheta \cos \gamma \sin \psi + \sin \vartheta \sin \gamma & -\sin \vartheta \cos \gamma \sin \psi - \cos \vartheta \sin \gamma & \cos \gamma \cos \psi
\end{bmatrix}
\]

The expressions for orientation angles take the form:

\[
\vartheta = \arctg \left( \frac{M_{bt_{12}}}{M_{bt_{11}}} \right); \quad \gamma = \arctg \left( \frac{M_{bt_{23}}}{M_{bt_{33}}} \right); \quad \psi = \arctg \left( \frac{M_{bt_{13}}}{\sqrt{M_{bs_{11}^2} + M_{bs_{12}^2}}} \right)
\]

b) Earth Model

Ellipsoid of rotation with the axis coinciding with the Earth rotation axis can be used as the Earth navigational model relative to the systems, where the method of inertial navigation is implemented:

a is the semi major axis, b is the semi minor axis, f is the flattening of the ellipsoid:

\[
e = \sqrt{f(2 - f)}; \quad e^2 = \frac{a^2 - b^2}{a^2}; \quad f = \frac{a - b}{a};
\]

WGS-84 international reference-ellipsoid with the following parameters is selected to be used in NMSW:

\[
a = 6\,378\,137 \text{ m} \\
f = 1/298.257223563 \\
e^2 = 6.6943799901413 \times 10^{-3} \text{ [m]}
\]
c) Algorithm for Compensating Instrument Errors

The model of compensating the errors of IMU sensors is used in the algorithm in the form of [15]:

\[
A_Z = M_{ZA} \cdot (K_A \cdot A_{RAW} - A_0)
\]

(5)

\[
\omega_s = M_{GZ} \cdot (\omega_Z + M_{\omega} \cdot A_Z + \omega_b + \omega_c)
\]

(6)

Where:

- \(A_Z\) is the vector of measured phantom acceleration used for velocity integration;
- \(A_{RAW}\) is the vector of accelerometer measurements;
- \(M_{ZA}\) the matrix of transition from accelerometer axes to an instrument trihedral;
- \(K_A\) is scale coefficients.
- \(A_0\) is a zero drift.
- \(\omega_s\) is a vector signal of platform control.
- \(\omega_Z\) is a design angular velocity of platform control.
- \(M_{GZ}\) is a compensation matrix of couplings and scale factor errors of gyroscope torquers.
- \(M_{\omega}\) is a matrix of dynamic drift coefficients;
- \(\omega_b\) is a zero drift vector (a drift constant component), and \(\omega_c\) is a case drift vector.

d) Velocity Integration and Navigational Calculations

To obtain relative velocity in the reference CS, the following differential equation is integrated:

\[
\ddot{V}_s = a_s - (\omega'_s + 2 \cdot U'_s) \times V'_s + g_s
\]

(7)

where:

- \(V'_s\) is a vector of relative velocity in the reference Coordinate system.
- \(a_s\) is a vector of measured phantom acceleration.
- \(\omega'_s\) is a vector of relative rate of a reference Coordinate system,
- \(U'_s\) is a vector of Earth angular velocity, and \(g_s\) is a gravity vector. The vector of relative velocity in the reference trihedral takes the form:

\[
\dot{V}'_s = (\dot{r}'_{s} - \ddot{r}'_{s} \cdot t_{m} - \dot{r}'_{s} \cdot t_{m-1})
\]

(8)

\[
\dot{r}'_{s} = (\omega'_s + 2 \cdot U'_s) \times V'_s
\]

Vector projections in the fixed CS are \([0, 0, -g_T]\).

The module of normal specific gravity [15]:

\[
g_T = g_e \cdot \left(1 + 0.005302 \sin^2 \varphi - 2 \frac{h}{a}\right)
\]

(9)

\[
g_e = 9.780319 \text{ m/s}^2
\]
projections of the components of relative rate of reference trihedral on their proper axes have the following form[15]:

\[
\begin{align*}
\omega'_{s1} &= -k_{s2} \cdot V'_{s2} + k_{s3} \cdot V'_{s1} \\
\omega'_{s2} &= k_{s1} \cdot V'_{s1} - k_{s3} \cdot V'_{s2} \\
\omega'_{s3} &= k_E \cdot V_E \cdot \tan \varphi
\end{align*}
\]

The curvature and pivoting of a flight path in the direction of the axes of the accompanying trihedral:

\[
\begin{align*}
k_{s1} &= k_E \cos^2 A + k_N \sin^2 A \\
k_{s2} &= k_E \sin^2 A + k_N \cos^2 A \\
k_{s3} &= (k_N - k_E) \cdot \sin A \cdot \cos A
\end{align*}
\]

Where A is the azimuth angle from north direction

\[
\begin{align*}
k_E &= \frac{1}{R_E} \\
k_N &= k_E \cdot \frac{1 - e^2 \cdot \sin^2 \varphi}{1 - e^2} \\
R_E &= \frac{a + h}{\sqrt{1 - e^2 \cdot \sin^2 \varphi}} \equiv \frac{a}{\sqrt{1 - e^2}} + h = N + h
\end{align*}
\]

The projections of Earth rate on the axes of the accompanying trihedral can be calculated by formulas [15]:

\[
\begin{align*}
U_{s1} &= -U \cdot \cos \varphi \cdot \sin A = U \cdot C_{13}^{Nav} \\
U_{s2} &= U \cdot \cos \varphi \cdot \cos A = U \cdot C_{23}^{Nav} \\
U_{s3} &= U \cdot \sin \varphi = U \cdot C_{33}^{Nav}
\end{align*}
\]

where: \( U = 15.04107 \,[\text{deg/h}] = 7.292115 \cdot 10^{-5} \, \text{l/s}. \)

Absolute angular rate of the reference CS:

\( \omega'_{r} = \omega'_{s} + \omega_{r} \)

The same vector in the axes of an instrument (fixed) trihedral:

\( \omega'_{k} = C_{i}^{ovi} \times \omega'_{r} \)

The components of a linear transition velocity of Earth rotation:

\[
\begin{align*}
V'_{s1} &= R_E \cdot U_{s2} \\
V'_{s2} &= -R_E \cdot U_{s1} \\
V'_{s3} &= 0
\end{align*}
\]

The vector of absolute linear velocity in the reference trihedral:

\( V'_{s} = V'_{s} + V_{r} \)
The components of a relative linear velocity in the geographic trihedral:
\[
\begin{align*}
V'_E &= V'_{s1} \cdot \cos A + V'_{r2} \cdot \sin A \\
V'_N &= -V'_{s1} \cdot \sin A + V'_{r2} \cdot \cos A \\
V'_U &= V'_{r3}
\end{align*}
\] (15)

**e) Calculation Algorithm of Geographic Coordinates**

Navigation algorithm is in the integration of \( C^{Nav} = \omega \times C^{Nav} \) Poisson equation and thence finding the matrix of direction cosines of navigation trihedral orientation, and hence – geographic coordinates: \( \varphi \) (latitude), \( \lambda \) (longitude), (azimuth) based on \( C^{Nav}(0) = f(\varphi_0, \lambda_0, A_0) \) initial value and \( \omega' \) angular rate of a navigation trihedral in the projections on its proper axes. Poisson equation is integrated on the basis of vector calculated value of \( \omega' \), relative rate, and the matrix of direction cosines of navigation trihedral can be determined by formulas:
\[
\begin{align*}
\hat{\omega}' &= 0 \quad \omega_1 \quad -\omega_2 \\
&\begin{bmatrix}
0 & \omega_1 & -\omega_2 \\
-\omega_1 & 0 & \omega_2 \\
\omega_2 & -\omega_1 & 0
\end{bmatrix}
\end{align*}
\] (16)

\[
C^{Nav}(\theta) = \begin{pmatrix}
\sin \varphi \cdot \cos \lambda \cdot \sin A - \sin \lambda \cdot \cos \lambda - \cos \varphi \cdot \sin A & \\
\cos \varphi \cdot \cos \lambda \cdot \sin A + \cos \lambda \cdot \sin A & -\cos \varphi \cdot \sin A \\
\cos \varphi \cdot \sin \lambda & \cos \varphi \cdot \cos \lambda
\end{pmatrix}
\] (17)

\[
\varphi = \arctan \frac{c_{33}}{\sqrt{c_{31}^2 + c_{32}^2}} \left[ -\frac{\pi}{2}, \frac{\pi}{2} \right]; \quad \lambda = \arctan \frac{c_{32}}{c_{31}} \left[ -\pi, \pi \right]; \quad A = \arctan \frac{-c_{13}}{c_{23}} \left[ -\pi, \pi \right]
\]

**4) In flight Alignment Error model**

We introduce the error model, which appropriately describes the INS behavior during IFA. We then substantiate its adequacy for consideration as a piece-wise constant system (PWCS) and then the simpler stripped observability matrix (SOM) can be used in the system observability analysis. The most suitable model for IFA, where the measured states are the velocity states, is the, so called psi-model [11,12,13,14]. This model is in the form:
\[
\begin{align*}
\begin{bmatrix}
\dot{\psi}_x \\
\dot{\psi}_y \\
\dot{\psi}_z
\end{bmatrix} &= 
\begin{bmatrix}
0 & -\tilde{\Omega}_U & 0 & -f_U & f_N \\
-\tilde{\Omega}_U & 0 & \tilde{\Omega}_E & f_U & 0 \\
\tilde{\Omega}_N & -\tilde{\Omega}_E & 0 & f_N & f_E
\end{bmatrix}
\begin{bmatrix}
\psi_x \\
\psi_y \\
\psi_z
\end{bmatrix} + 
\begin{bmatrix}
\nabla_x \\
\nabla_y \\
\nabla_z
\end{bmatrix}
\end{align*}
\] (18)
Where $V$ and $\psi$ represent the velocity error and attitude error states, respectively; $\nabla$ and $\epsilon$ represent, respectively, the generalized accelerometer error and the generalized gyro drift rate; $f_E$, $f_N$, and $f_U$ represent, respectively, the East, North, and Up components of the specific force sensed by INS respective accelerometers. $\lambda$ and $\varphi$ represent, respectively, the longitude and the latitude of the INS; and $\Omega$ is the earth rate.

\[
\dot{\bar{\Omega}} = \begin{bmatrix} 0 & -\bar{\Omega}_U & -\bar{\Omega}_N \\ -\bar{\Omega}_U & 0 & -\bar{\Omega}_E \\ -\bar{\Omega}_N & -\bar{\Omega}_E & 0 \end{bmatrix} ; 
\frac{d}{dt} \begin{bmatrix} 0 & \Omega_U & -\Omega_N \\ -\Omega_U & 0 & \Omega_E \\ \Omega_N & -\Omega_E & 0 \end{bmatrix}; \quad F_j = \begin{bmatrix} 0 & -f_U & f_N \\ f_U & 0 & -f_E \\ -f_N & f_E & 0 \end{bmatrix}
\]  

(19)

Consequently, the carrier trajectory during the IFA maneuver can be characterized by a sequence of segments taken from table (1). The elements $f_E$, $f_N$, and $f_U$ for each segment $j$ are the entries of the sub matrices $F_j$

<table>
<thead>
<tr>
<th>Number of Segment (j)</th>
<th>Maneuver Characteristics</th>
<th>$f_E$</th>
<th>$f_N$</th>
<th>$f_U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>straight flight</td>
<td>0</td>
<td>0</td>
<td>-g</td>
</tr>
<tr>
<td>2</td>
<td>maneuver with north acceleration</td>
<td>0</td>
<td>$f_N$</td>
<td>-g</td>
</tr>
<tr>
<td>3</td>
<td>maneuver with east acceleration</td>
<td>$f_E$</td>
<td>0</td>
<td>-g</td>
</tr>
<tr>
<td>4</td>
<td>maneuver with horizontal acceleration</td>
<td>$f_E$</td>
<td>$f_N$</td>
<td>-g</td>
</tr>
<tr>
<td>5</td>
<td>Pull-up or dive</td>
<td>0</td>
<td>0</td>
<td>- $f_U$</td>
</tr>
</tbody>
</table>

Table (1) Trajectory segmentation

The observability analysis of the INS at segment $j$, after having gone through segments $1, 2, \ldots, j-1$ from table (1), the total observability matrix (TOM) $\tilde{Q}(j)$ is constructed as follows:

\[
\tilde{Q}(j) = \begin{bmatrix} \bar{Q}_1 \\ \bar{Q}_2 e^{\Delta_1} \\ \bar{Q}_j e^{\Delta_{j-1}} \end{bmatrix} M e^{\Delta_1}, \quad 1 \leq i \leq j
\]  

(20)

Where

\[
\bar{Q}_i^T = \begin{bmatrix} H_i^T & (H_i A_i^2)^T & K (H_i A_i^{n+1})^T \end{bmatrix}^T, \quad 1 \leq i \leq j
\]  

(21)

$\Delta_i$ is the time span of segment $i$.

Consequently, instead of using the TOM of Equation (5.3.1) we may use $\tilde{Q}_i(j)$ stripped observability matrix (SOM), which simplifies the observability analysis considerably. The SOM $\tilde{Q}_i(j)$ is constructed as[14]:
\[ \tilde{Q}_s(j) = \begin{bmatrix} \tilde{Q}_1 \\ \tilde{Q}_2 \\ M \\ \tilde{Q}_s \end{bmatrix} \]  

(22)

If \( \text{null}(\tilde{Q}_j) \subset \text{null}(A_j) \), \( \forall \ j = 1, 2 \ldots r \)

Then, \( \text{null}(\tilde{Q}_j) = \text{null}(\tilde{Q}_s) \); \( \text{rank}(\tilde{Q}_j) = \text{rank}(\tilde{Q}_s) \)

5) Simulation and results

From the trajectory data, under the assumption of the heading angle 60°, constant velocity 400[m/s], initial misalignment angles is chosen as 1°, measuring error of velocity is 0.5 [m/s], and the local latitude of SINS place is 30°. Also, the trajectory data, under the turn maneuver with maximum variation of heading angle 30°.

Table (2) shows the different convergence rates of \( \psi_x, \psi_y, \) and \( \psi_z \) for 10-states and 12-states in the cases of straight level flight, and turn level maneuver. Also, figures (3, 4), and figures (5, 6) shows the result of azimuth estimated error, for 10-states and 12-states in the above two cases and figures (7,8,9,10) shows some of the missile flight parameters.

<table>
<thead>
<tr>
<th>Case</th>
<th>( \Delta \psi_x ) [sec]</th>
<th>( \Delta \psi_y ) [sec]</th>
<th>( \Delta \psi_z ) [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-states (2-channel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Straight level-flight</td>
<td>-6.3733</td>
<td>-29.1415</td>
<td>-211.3699</td>
</tr>
<tr>
<td>turn level maneuver</td>
<td>0.8430</td>
<td>-0.5898</td>
<td>-1.7385</td>
</tr>
<tr>
<td>12-states (3-channel)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Straight level-flight</td>
<td>-6.4982</td>
<td>-28.8416</td>
<td>-231.7737</td>
</tr>
<tr>
<td>turn level maneuver</td>
<td>-0.2757</td>
<td>0.3022</td>
<td>-3.7159</td>
</tr>
</tbody>
</table>

Table (2) Estimation error of drift misalignment angles \( \psi_x, \psi_y, \) and \( \psi_z \) for 10 and 12-states
Conclusion

The Launch vehicle will be controlled by a small on-board digital computer with navigation data from an inertial system. In this paper we focused on the Navigation mathematical Software (NMSW) which performs the following functions such as INS initial alignment, Solution of navigation problems: inertial velocity and coordinate reckoning in addition to inertial sensors calibration. The methods of testing the program are described, and a typical trajectory is given based on the hard ware in the loop missile flight simulation. The attention was focused on the derivation and programming of NMSW algorithm for SINS wanders azimuth mechanization. Based on the trajectory simulator data from the missile equation of motions the process of simulation and computation is highlighted.
Also, we introduced the error model, which appropriately describes the INS behavior during IFA. We then substantiated its adequacy for consideration as a piece-wise constant system (PWCS). Using PWCS we can simplify in-flight alignment observability analysis. The simulation results showed the effect of the variation of initial large azimuth misalignment angle for two-channel and three channels including the un-damped channel by using the extended Kalman filter (EKF).

References


[14] Goshen-Meskin, Bar-Itzback Israel Aircraft and Institute of Technology “Observability}
|---|