

Diseño de software aviónico con integración de MISRA con VxWorks en aplicaciones de misil-balístico

Verónica Quintero-Rosas¹, Coen O'Kelly², Atalo Machado¹

¹Departamento de Sistemas y Computación, Tecnológico Nacional de México/ Instituto Tecnológico de Mexicali, Av. Tecnológico s/n, Col. Elías Calles, Mexicali BC, 21376

² Honeywell 110 Union St Suite 400, Seattle, WA 98101

Resumen

La aviónica engloba sistemas de navegación, comunicación, radar, control y monitoreo de vuelo, que permiten a los pilotos operar la aeronave de manera segura y eficiente. Un misil balístico es un proyectil autopropulsado diseñado para seguir una trayectoria balística, lo que significa que sigue una trayectoria parabólica determinada y sigue una inercia una vez que se lanza. A diferencia de otros misiles, estos no se mantienen propulsados durante todo su vuelo ya que utilizan un motor de cohetes para acelerar a velocidades altas y luego siguen una trayectoria parabólica libre bajo la influencia de la gravedad. Los sistemas de control y navegación de un misil son responsables de calcular y ajustar continuamente la trayectoria del misil para que alcance su objetivo con precisión. La programación de la trayectoria generalmente se realiza utilizando un conjunto de tecnologías y software específicos, en nuestro caso se utilizará VxWorks con Misra y nos enfocaremos a una programación del Sistema de Navegación Inercial (INS) utilizando giroscopios y acelerómetros para rastrear los cambios en la velocidad y la posición del misil. La información recopilada por los sensores se utiliza para calcular la trayectoria actual del misil y realizar correcciones según sea necesario. Es importante destacar que la programación de la trayectoria de un misil está altamente clasificada y sujeta a regulaciones gubernamentales y acuerdos internacionales, especialmente en el caso de misiles balísticos y misiles de largo alcance, debido a su potencial impacto en la seguridad global, por ello este artículo solo tiene un carácter educativo en el desarrollo de software ya que los algoritmos para programar son dinámicos para seguir ciertas acciones

Abstract

Avionics encompass navigation, communication, radar, flight control and monitoring systems, which allow pilots to operate the aircraft safely and efficiently. A ballistic missile is a self-propelled projectile designed to follow a ballistic trajectory, meaning it follows a set parabolic trajectory and inertia once it is launched. Unlike other missiles, these do not remain propelled throughout their flight as they use a rocket motor to accelerate to high speeds and then follow a free parabolic trajectory under the influence of gravity. A missile's control and navigation systems are responsible for continually calculating and adjusting the missile's trajectory so that it hits its target accurately. Trajectory programming is generally done using a set of specific technologies and software, in our case we will use VxWorks with Misra and we will focus on Inertial Navigation Systems (INS) programming using gyroscopes and accelerometers to track changes in speed and the position of the missile. The information collected by the sensors is used to calculate the missile's current trajectory and make corrections as necessary. It is important to note that missile trajectory programming is highly classified and subject to government regulations and international agreements, especially in the case of ballistic missiles and long-range missiles, due to its potential impact on global security, which is why this The article only has an educational nature in software development since the algorithms to be programmed are dynamic to follow certain actions.

Palabras Clave: Misra, software avionico, vxworks, aplicaciones militares

Keywords: Misra, avionic software, vxworks, military applications

1. INTRODUCCIÓN

Existen técnicas efectivas para evaluar la confiabilidad de un sistema de software. Esta técnica [1] es de tipo analítica, con requerimientos de programa mediante certificados matemáticos, de ajuste a requisitos

formalmente especificados y criterios respecto a datos de entrada- esperados con datos esperados de recibo. Sin olvidar los modelos matemáticos de corrección, para garantizar que un sistema de software cumpla con sus especificaciones teóricas, y de verificación de curso.

Las variables externas que modifican el curso de los algoritmos [2] en este caso sensores, son los responsables del recalcu en tiempo real de corrección de rumbo o continuación del mismo. Esto puede afectar a las diferentes etapas del misil como lo es Propulsión inicial; donde utiliza un motor de cohetes para acelerar rápidamente a velocidades hipersónicas. Este motor se enciende en el momento del lanzamiento y proporciona una aceleración masiva. El Apogeo es, una vez que el motor se apaga, el misil alcanza su punto más alto en la trayectoria, conocido como apogeo. En este punto, deja de ascender y comienza a descender por la gravedad. Otra etapa es la trayectoria libre, que es durante la fase de apogeo, el misil sigue una trayectoria libre, que es una curva en forma de parábola. La gravedad actúa sobre el misil, haciendo que descienda hacia su objetivo. Por último, la velocidad extrema, en esta etapa los misiles balísticos [3] pueden alcanzar velocidades extremadamente altas durante la fase de propulsión, lo que los hace difíciles de interceptar o detener.

Los misiles balísticos se utilizan en una variedad de aplicaciones, desde misiles balísticos intercontinentales (ICBM) [7], diseñados para transportar ojivas nucleares a largas distancias hasta misiles balísticos de corto alcance utilizados en conflictos militares convencionales. Estos misiles son conocidos por su velocidad y alcance, lo que los hace una amenaza seria en contextos militares.

Es importante señalar que, debido a su capacidad para llevar ojivas nucleares y alcanzar objetivos a largas distancias, los misiles balísticos han sido objeto de regulaciones internacionales y tratados de desarme, como el Tratado de Reducción de Armas Estratégicas (START) y el Tratado sobre la No Proliferación de Armas Nucleares (TNP) [8]. Estos tratados tienen como objetivo controlar y limitar la proliferación de misiles balísticos con ojivas nucleares.

Un misil balístico [9], como los misiles balísticos intercontinentales (ICBM) y otros misiles de largo alcance, generalmente pasan por varias fases distintas durante su vuelo. Estas fases se diseñan cuidadosamente para lograr el objetivo final del misil. Las fases típicas de un ICBM son las siguientes:

Fase de Impulso: En esta fase, el motor del cohete se enciende en el momento del lanzamiento y proporciona una aceleración masiva al misil. El objetivo es alcanzar la velocidad y altitud deseadas. Esta es la fase más corta del vuelo y termina cuando se agota el combustible del motor.

Fase de Ascenso: Después de la fase de impulso, el misil continúa ascendiendo y se acerca a su punto más alto en la trayectoria, conocido como apogeo. Durante esta fase, se apagan los motores y el misil sigue una trayectoria balística libre bajo la influencia de la gravedad.

Fase de Descenso: En esta fase el misil desciende hacia su objetivo en la Tierra. Durante esta fase, la velocidad del misil disminuye y se dirige hacia el objetivo.

Fase de Reentrada: En esta etapa, el misil vuelve a entrar en la atmósfera terrestre desde el espacio exterior. La ojiva de la cabeza de combate está protegida por un escudo térmico que la protege del calor extremo generado durante la reentrada. Esta fase es crítica, ya que la ojiva debe resistir las altas temperaturas y las fuerzas g durante el descenso.

Fase de Terminal: En esta etapa, el misil se encuentra cerca de su objetivo y realiza maniobras finales para dirigirse con precisión hacia su objetivo. Los sistemas de guía y control ajustan la trayectoria para alcanzar el punto exacto de impacto. No todos los misiles siguen exactamente estas fases, y algunas misiones de misiles pueden tener características específicas diferentes. En contextos militares, los misiles balísticos pueden ser objeto de contramedidas y sistemas de defensa antimisiles diseñados para interceptarlos durante sus diferentes fases. Por lo tanto, las fases de un misil y su capacidad de eludir las defensas enemigas son elementos críticos en su diseño y operación.

La programación de la trayectoria de un misil es, una tarea compleja que implica una variedad de sistemas y software especializado. Los sistemas de control y navegación de un misil son responsables de calcular y ajustar continuamente la trayectoria del misil para que alcance su objetivo con precisión. La programación de la trayectoria generalmente se realiza utilizando un conjunto de tecnologías y software específicos, que pueden incluir:

Sistemas de Navegación Inercial: Utilizan giroscopios y acelerómetros para rastrear los cambios en la velocidad y la posición del misil. La información recopilada por los sensores inerciales se utiliza para calcular la trayectoria actual del misil y realizar correcciones según sea necesario.

Sistemas de Guía: Los sistemas de guía son responsables de dirigir el misil hacia su objetivo. Estos sistemas pueden utilizar sensores como radares, sistemas de seguimiento visual, infrarrojos u otros sensores para detectar y seguir el objetivo. Los datos recopilados por los sensores se utilizan para realizar ajustes en la trayectoria del misil.

Computadora a Bordo: Las computadoras de a bordo del misil son responsables de procesar datos de navegación, datos del objetivo y otros parámetros. Utilizan algoritmos y software especializados para calcular y ajustar la trayectoria del misil en tiempo real.

Software de Control de Vuelo: El software de control de vuelo del misil ejecuta los algoritmos que controlan los actuadores del misil, como aletas de control o chorros de dirección, para realizar las maniobras necesarias y ajustar la trayectoria.

Sistemas de Propulsión: El motor del cohete del misil también desempeña un papel importante en la programación de la trayectoria. El control de la potencia y la dirección del motor se utiliza para ajustar la velocidad y la orientación del misil a lo largo de su vuelo.

La programación de la trayectoria de un misil es un proceso continuo y altamente dinámico. A medida que el misil se acerca a su objetivo, se realizan ajustes finos en la trayectoria para asegurar que impacte con precisión. La combinación de estos sistemas y tecnologías permite que los misiles alcancen objetivos en movimiento o fijos con alta precisión.

2. DESARROLLO

Una defensa contra misiles balísticos depende en gran medida en el software que lo controla. Por tanto, el desarrollo de software, es un factor crítico para alcanzar los objetivos para los cuales está diseñado el sistema.

El desarrollo de software es un proceso de distintas fases conceptuales, incluyen planificación, diseño, implementación, pruebas y depuración. La primera área crucial del desarrollo de software es la planificación: el desarrollador necesita determinar qué funciones el software realizara y visualizara, además de las diferentes situaciones a las que el software debe responder o decidir qué acción se debe tomar y en qué momento, lo que lo hace crítico y complejo. En las pruebas a pequeña escala [4], a menudo se utilizan simuladores para crear posibles escenarios de ataque. Aunque los simuladores no puedan reproducir las "firmas" o parámetros de los fenómenos físicos asociados con eventos inesperados, tienen suficiente rapidez y precisión para probar la capacidad de respuesta del sistema de defensa. De esta manera se analiza mediante simulación que firma o parámetro se elige simular; ya que todo debe basarse en suposiciones. Las elecciones del evento a simular están restringidas ya que deben estar "preprocesados" [7] antes de ser alimentados en el sistema de defensa. Como resultado de utilizar datos preprocesados, entonces no habrá sorpresas durante, la simulación. Lo cual es una limitante crítica al probar cómo reacciona el sistema a circunstancias no previstas por los programadores, y por lo tanto el realismo de la prueba está disminuido. Existen simuladores que pueden modelar entornos más realistas, una mayor velocidad etc, pero esto no ayuda a simular datos físicos o la comprensión empírica, ya que todo está basado en la cuestión teórica, ya que se basa en el supuesto de que los responsables de la simulación pueden predecir y reproducir en forma electrónica las tácticas a las que un enemigo podría recurrir.

VxWoks

VxWoks [14] es un sistema operativo multitarea en tiempo real orientado a sistemas empotrados. Proporciona multitud de servicios como gestión de memoria, multitarea, sistemas de ficheros, redes, etc. y que está disponible para múltiples arquitecturas de procesadores y computadores industriales. Es un sistema operativo escalable, muy fiable y ampliamente utilizado en sectores muy especializados como industria aeroespacial, sistemas médicos, militar, comunicaciones, etc.

MISRA

Misra [13] establece estándares y directrices para el desarrollo de software donde se requieren alta confiabilidad y seguridad en el software. Los estándares MISRA se utilizan ampliamente en la industria automotriz, la aeroespacial y otras áreas donde la seguridad y la fiabilidad son fundamentales. Los estándares MISRA se centran en el desarrollo de software en lenguaje C y C++, y están diseñados para abordar los desafíos específicos relacionados con la seguridad, la integridad y la confiabilidad del software en sistemas embebidos y críticos para la seguridad. MISRA proporciona reglas detalladas y pautas de codificación para garantizar que el software se desarrolle de manera consistente y siga las mejores prácticas, también se centra en minimizar riesgos, evitar comportamientos indefinidos y garantizar la seguridad y la fiabilidad del software. Además de controlar la complejidad del código, legibilidad, verificación, validación además de su documentación para su mantenimiento y comprensión del código.

La manera de verificar datos de sensor se estipula con el siguiente código.

```
#include <vxWorks.h>
#include <stdio.h>
#include <taskLib.h>
float readSensor() {
    return 20.2;
```

```

}
void workSensor() {
    float lectura;
    while (1) {
        lectura = readSensor();
        printf("Lectura del sensor: %.2f\n", lec);
    }
}
int main() {
    if (taskSpawn("workSensor", 100, 0, 10000, (FUNCPTR)workSensor, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0) == ERROR) {
        printf("Error sensor.\n");
        return -1;
    }
}

```

Para verificar la función de navegación y seguimiento de objetivo se utiliza el siguiente algoritmo donde se llama recursivamente la función de navegación, calcular el ángulo entre dos puntos, Función auxiliar para calcular la velocidad en función de la distancia etc.

```

Funcion Navegacion (objetivoX, objetivoY, posActualX, posActualY):
    dis = CalDistancia(objetivoX, objetivoY, posActualX, posActualY)
    if dis < umbral:
        printf( "Objetivo en la mira");
    else
        angulo = CalÁngulo(objetivoX, objetivoY, posActualX,
posActualY)
        velocidad = CalVelocidad(distancia)
        dirección = angulo
        posActualX = posActualX + velocidad * cos(dirección)
        posiciónActualY = posActualY + velocidad * sin(dirección)
        Navegacion(objetivoX, objetivoY, posActualX, posActualY)
Funcion CalDistancia(x1, y1, x2, y2):
    distancia = sqrt((x2 - x1)^2 + (y2 - y1)^2)
    Devolver distancia
Función CalÁngulo(x1, y1, x2, y2):
    angulo = atan2(y2 - y1, x2 - x1)
    Devolver ángulo
Función CalVelocidad(distancia):
    velocidad = distancia * factorDeVelocidad
Navegación(objetivoX, objetivoY, posiciónActualX, posiciónActualY)

```

Giroscopio y su programación de recalcu de trayectoria

Un giroscopio es un dispositivo para medir o mantener la orientación y la dirección en sistemas de navegación. Funciona en base al principio del momento angular y la conservación del movimiento angular [6]. El giroscopio se basa en el principio de que un objeto en movimiento tiende a mantener su eje de rotación en su dirección original. Cuando un giroscopio está en movimiento o sujeto a cambios en la orientación, tiende a resistir esos cambios y mantener su orientación original. Consta de un rotor giratorio que gira a alta velocidad. La conservación del momento angular del rotor ayuda a mantener su orientación original en el espacio. Los

giroscopios de anillo láser (que utilizan interferencia de luz para medir la rotación) es el utilizado en este trabajo, debido a su capacidad de ajuste y precisión. A continuación, se muestra el pseudocódigo de variación del giroscopio

```

const int MPU6050_addr = 0x68;
int16_t gyroX, gyroY, gyroZ;
void setup() {
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x6B);
  Wire.write(0); // wake up sensor
  Wire.endTransmission();
  Serial.begin(9600);
}void loop() {
  Wire.beginTransmission(MPU6050_addr);
  Wire.write(0x43); // velocidad angular en el eje X
  Wire.endTransmission();
  Wire.requestFrom(MPU6050_addr, 6);
  if (Wire.available() >= 6) {
    gyroX = Wire.read() << 8 | Wire.read();
    gyroY = Wire.read() << 8 | Wire.read();
    gyroZ = Wire.read() << 8 | Wire.read();
  } Serial.print("Velocidad angular en X: ");
  Serial.println(gyroX / 131.0); // Factor de escala para convertir a grados
por segundo
  Serial.print("Velocidad angular en Y: ");
  Serial.println(gyroY / 131.0);
  Serial.print("Velocidad angular en Z: ");
  Serial.println(gyroZ / 131.0);
  delay(1000);
}

```

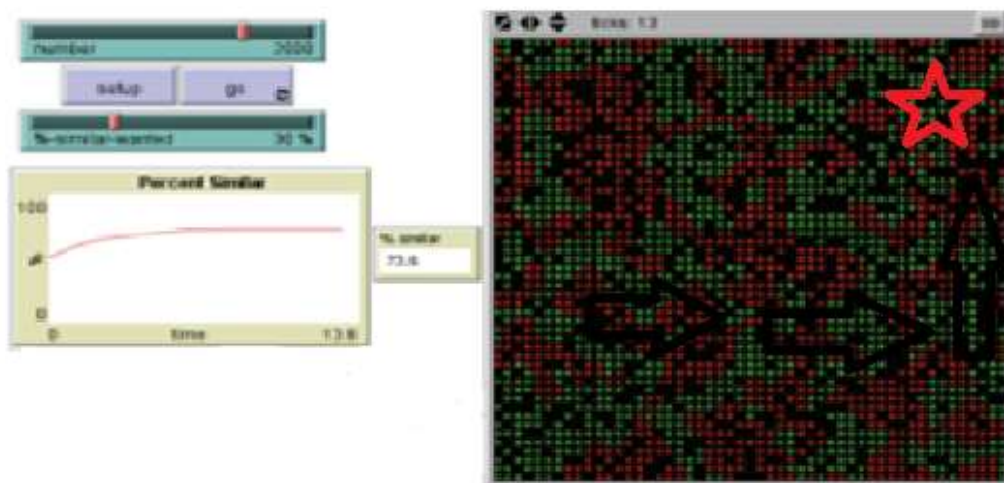


Figura 1. Matriz de simulación con posibles puntos de trayectorias del misil, la estrella es el objetivo y la gráfica muestra la estabilidad de lo programado (negro) con el comportamiento de posiciones (rojo).

3. CONCLUSIONES

Esta es la primera investigación que se hace de software balístico utilizando VxWorks con los lineamientos de MISRA, en vinculación con un ingeniero de software militar de Honeywell. Se necesita hacer más investigación y más pruebas, ya que la simulación está realizada en plataforma cerrada a la cual no se tiene acceso por cuestiones de seguridad; sin embargo, esta primer fase arrojó menor número de escenarios posibles (rojo) a donde el misil podría dirigirse lo cual es positivo, sin embargo, no tenemos los parámetros físicos con los cuales se llevó a cabo dicha simulación, lo cual nos limita a saber cómo mejorar los programas.

Cabe señalar que esta primera fase de la investigación nuestro estudiante aprendió Vxworks y todos los lineamientos y reglas de MISRA, ya que son herramientas que son utilizadas mediante licencias especializadas y usos limitados a empresas específicas. Por tanto, nuestra institución no cuenta con dichas licencias y certificaciones por lo que Honeywell proporciona el uso en sus servidores para la programación y simulación.

Esta es una investigación en proceso, se espera que la siguiente parte de esta investigación este estructurada con especificaciones de simulación más abiertas, para poder hacer el análisis correspondiente y optimizar el código conforme a resultados del giroscopio y su objetivo a seguir por el misil.

REFERENCIAS

- [1] B. E. Logan and G. G. Xie, "Automating distributed firewalls: A case for software defined tactical networks," in IEEE Military Communications Conference, ser. MILCOM 2019. Norfolk, VA, USA: IEEE, Nov. 2019
- [2] Hansson, J. Nilsson, P. Lubkowski, "A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks," in International Conference on Military Communications and Information Systems, ser. ICMCIS
- [3] I. Brown, J. H. Kim, and M. Gerla, "An integrated softwaredefined battlefield network testbed for tactical scenario emulation," in IEEE Military Communications Conference, ser. MILCOM 2019.
- [4] Norfolk, VA, USA: IEEE, Nov. 2019, pp. 373–378. Cai, Z., Gu, Y., & Zhong, W. (2001). A new approach of computing Floquet transition matrix. *Computers & Structures*, 79(6), 631-635.
- [5] Chen, G., & Yang, Y. (2016). New stability conditions for a class of linear time-varying systems. *Automatica*, 71, 342-347.
- [6] Duc, L., Ilchmann, A., Siegmund, S., & Taraba, P. (2006). On stability of linear time-varying second-order differential equations. *Quarterly of applied mathematics*, 64(1), 137-151.
- [7] Floquet, G. (1883). Sur les équations différentielles linéaires à coefficients périodiques. In *Annales scientifiques de l'École normale supérieure* (Vol. 12, pp. 47-88).
- [8] J. Loevenich, R. R. F. Lopes, P. H. Rettore, S. M. Eswarappa, and P. Sevenich, "Maximizing the probability of message delivery over ever-changing communication scenarios in tactical networks," *IEEE Networking Letter*, pp. 1–5, March 2021,
- [9] R. R. F. Lopes, P. H. Balaraju, P. H. Rettore, and P. Sevenich, "Queuing over ever-changing communication scenarios in tactical networks," *IEEE Transactions on Mobile Computing*, pp. 1–15, June 2020.
- [10] P. H. Balaraju, P. H. Rettore, R. R. F. Lopes, S. M. Eswarappa, and J. Loevenich, "Dynamic adaptation of the user data flow to the changing data rates in VHF networks: An exploratory study," in 11th IEEE International Conference on Network of the Future (NoF), Bordeaux, France, Oct 2020, pp. 64–72.
- [11] R. R. F. Lopes, P. H. Balaraju, and P. Sevenich, "Creating and handling ever-changing communication scenarios in tactical networks," in 15th International Conference on the Design of Reliable Communication Networks (DRCN), Coimbra, Portugal, March 2019, pp. 67–74.
- [12] R. R. F. Lopes, J. Loevenich, P. H. L. Rettore, S. M. Eswarappa, and P. Sevenich, "Quantizing radio link data rates to create ever-changing

network conditions in tactical networks,” IEEE Access, vol. 8, pp. 188 015–188 035, 2020.

[13] Roberto Bagnara, MISRA C:2023: Guidelines for the use of the C language in critical systems

[14] Miriam Tenow Vxworks ISBN-10 : 6133565578

Correo de autor de correspondencia: veronicaquintero@itmexicali.edu.mx