



UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO INGENIERÍA MECÁNICA



**ESTABLECIMIENTO DE PROTOCOLO DE COMUNICACIÓN ENTRE X-PLANE
Y MICROPROCESADORES EXTERNOS**

POR

Germán Adolfo Quijada Arriagada

Profesor guía:

Bernardo Andrés Hernández Vicente

14 de mayo de 2023

Índice general

1. Planteamiento del problema	2
1.1. Contexto y justificación	2
1.2. Software utilizado	3
1.3. Otras soluciones	4
2. Objetivos	5
2.1. Objetivo general	5
2.2. Objetivos específicos	5
3. Condiciones de diseño	6
3.1. Simulador	6
3.2. Protocolo de comunicación	7
3.3. Documentación	7
4. Metodología de trabajo	8
4.1. Simulador	8
4.2. Protocolo de comunicación	9

Capítulo 1

Planteamiento del problema

1.1. Contexto y justificación

Los programas de ingeniería aeroespacial imparten aprendizaje introductorio de los procedimientos necesarios para ejecutar un vuelo en lo que es el pregrado, y abren las puertas a la posterior investigación de temas relacionados con la aeronáutica. Ambas actividades requieren de equipo adecuado que permita realizar simulaciones que sustenten estos estudios [1], con el objetivo de lograr familiarizar al estudiante con los diferentes sistemas y fenómenos que surgen durante el vuelo, o verificar las hipótesis de una investigación con cierto grado de realismo y confianza.

El trabajar con simulaciones abre la posibilidad de acceder o modificar arbitrariamente las variables de la aeronave y el entorno, utilizando interfaces que extraen de la simulación el estado de los sistemas involucrados y lo exportan a otros medios en los que sea más fácil hacer uso de los datos, sea visualizarlos en tiempo real en un panel de instrumentos con indicadores, el registro de la condición de la aeronave durante el vuelo para posterior evaluación de rendimiento, o la modificación directa de la simulación para estudiar condiciones específicas, entre otros.

Sin embargo, existen barreras tecnológicas que ralentizan el uso de un simulador de vuelo ya que el usuario debe en primera instancia familiarizarse con el funcionamiento interno del software y contar con el hardware adecuado que ejecute la simulación. Este documento trata con el caso específico de lograr comunicar el estado de la simulación a un hardware externo y que este pueda enviar comandos de vuelta que sean interpretados por el software,

estableciendo una interfaz dedicada para ello con el principal objetivo de facilitar el uso de una herramienta así.

El uso de hardware externo que no sea la misma computadora que está realizando la simulación es común en el círculo de entusiastas de la aviación o pilotos, la necesidad de crear experiencias realistas lleva a la implementación de controles e instrumentación personalizada conectada a microcontroladores, los que deben interactuar con el software por medio de alguna interfaz. Además, y sobre todo en un contexto de investigación donde se deban realizar múltiples pruebas iterativas, el que se trabaje con equipos externos evita que se utilicen recursos que pudiese requerir la simulación principal y da protección frente alguna modificación del software de manera permanente por error, así el equipo que alberga el simulador se mantiene íntegro asegurando su disponibilidad para ser utilizado en un futuro con otro propósito.

1.2. Software utilizado

Las principales soluciones para comunicar simuladores de vuelo con hardware externo tipo microprocesadores Arduino son SimVimX [4] y MobiFlight [5], estos programas están diseñados principalmente para entusiastas que construyen recreaciones de los paneles de un avión con controles e instrumentos personalizados, y facilitan la integración de estos mismos con el simulador gracias a que proporcionan manuales para esto e interfaces gráficas que ayudan a programar todo.

Debido al enfoque que tienen SimVimX y MobiFlight, las funciones que puede cumplir el microcontrolador están limitadas alrededor del problema que quieren solucionar de hacer de interfaz entre instrumentos y el simulador, y no permiten hacer mucho más sin tener que modificar código por lo que no son muy útiles para otros fines. Se podría decir que se trata de soluciones de alto nivel que en si ya dependen de otro software de más bajo nivel.

Flight Simulator Universal Inter-Process Communication [6] o FSUIPC es un software que accede a la memoria interna de un simulador en tiempo real (específicamente Microsoft Flight Simulator) y permite a otros programas modificarla de una manera estandarizada, muchos plugins existentes dependen de esta interfaz para funcionar incluyendo SimVimX y MobiFlight. Además, el protocolo que utiliza FSUIPC puede ser considerado como un estándar ya que existen adaptaciones de este para X-Plane como son XPUIPC [7] o XConnect [8] que lo imitan, lo que posibilita que plugins hechos con Microsoft Flight Simulator en mente también funcionen en X-Plane.

Programar software que aproveche FSUIPC (y por lo tanto XPUIPC para X-Plane) puede realizarse en lenguajes como C, Java o Python entre muchos otros, lo que permite aprovechar las ventajas de cada lenguaje en específico como lo son la velocidad o el strong-typing de C, o Python y su gran cantidad de librerías de terceros donde son de especial interés aquellas que conectan a Arduino o facilitan la comunicación por IP. La referencia del software documentada en manuales y existen ejemplos incluidos en la descarga del kit de desarrollo.

1.3. Otras soluciones

X-Plane Connect o XPC es otra interfaz diseñada por la NASA que cumple funciones similares a FSUIPC, pero específica para X-Plane sin posibilidad de adaptarla a otros simuladores. Lo interesante de esta alternativa es que incluye una librería para programar en MATLAB y por consiguiente Simulink lo que facilita el análisis de sistemas de control.

SimConnect [9] desarrollado por Microsoft es una interfaz de bajo nivel como XPC o FSUIPC diseñada exclusivamente para Microsoft Flight Simulator 2020 también con características similares a las demás librerías.

Capítulo 2

Objetivos

2.1. Objetivo general

Reconfigurar simulador de vuelo, luego con el simulador funcionando adecuadamente establecer un protocolo de comunicación entre el software del simulador y hardware externo, finalmente preparar material que facilite el uso del simulador en el futuro.

2.2. Objetivos específicos

1. Reconfigurar simulador de vuelo con el objetivo de facilitar su uso y agilizar en el futuro la instalación de nuevo software. Se registrará y comparará el hardware actual del simulador con el hardware nuevo disponible en el laboratorio para evaluar la implementación de una nueva configuración de acuerdo con los objetivos.

2. Establecer protocolo de comunicación entre software del simulador y equipos externos, primero estableciendo un canal de comunicación por IP generalizado, luego preparar con librerías existentes la extracción y manipulación de variables del simulador para finalmente combinar el canal y la implementación de la librería en un protocolo personalizado.

3. Facilitar uso del simulador preparando manuales instructivos referente a la configuración de hardware y software del simulador, sus capacidades e instrucciones de uso para vuelo general y el trabajo con el protocolo de comunicación con dispositivos externos.

Capítulo 3

Condiciones de diseño

3.1. Simulador

Se puede notar del objetivo específico de reconfigurar el simulador que una nueva implementación de hardware/software depende de lo que está actualmente disponible en el laboratorio, aunque de ser necesario adquirir equipo como cables o adaptadores estos no representan un costo elevado por lo que se consideran en el diseño. Actualmente para encender e iniciar el simulador se deben realizar ciertos pasos los que para alguien no introducido al equipo pueden resultar confusos, el diseño de una nueva implementación debe agilizar este proceso reduciendo los pasos necesarios lo que implica también simplificar la disposición del hardware y automatización por medio de software.

La nueva configuración debe ser simple para que usuarios en el futuro puedan ajustar fácilmente el simulador a nuevas necesidades sin tener que reestructurar todo, por lo que la disposición interna de archivos del sistema operativo debe ser de libre acceso para las modificaciones que se vayan a hacer. Sin embargo, el acceso completo va a llevar a que ocurran problemas en el software que se deben reparar, la nueva configuración debe tener esto en cuenta permitiendo una rápida restauración de los programas del simulador y que a la vez permita la experimentación con nuevo software.

Ya adentrándose en las condiciones de diseño específicas del software, se va a trabajar con alguna variante de Windows sea esta Windows 10 u 11 dependiendo de las licencias con las que cuente la universidad. De la misma manera el software de simulación es el que proporciona la universidad como por ejemplo X-Plane 9 que ya es utilizado en el ramo Mecánica de Vuelo.

Teniendo en cuenta la condición de diseño previa de facilitar la libre experimentación también debe existir un espacio donde instalar nuevo software, el que lo más probable será compatible con ciertos sistemas operativos como Windows o que se conectará por alguna interfaz ya existente a FSUIPC si se trata de un simulador por lo tanto la nueva configuración estará diseñada para aumentar la compatibilidad con estos sistemas.

3.2. Protocolo de comunicación

En la misma línea de aumentar la compatibilidad, el nuevo código que se vaya a escribir para el protocolo de comunicación utilizara estándares establecidos que usen el común de los simuladores, y la implementación en el lado del hardware externo también será en un lenguaje que compartan los microprocesadores o microcontroladores como Arduino y Raspberry con mínima modificación.

Mantener la complejidad al mínimo en el código es clave, pues se está buscando que esta interfaz sea utilizada por más personas que no necesariamente están familiarizadas con la programación y que las que si lo están sean capaces de leer y mejorar el programa.

3.3. Documentación

Respecto a la documentación esta será en formato tipo manual y abarcará principalmente el uso del simulador para misiones comúnmente realizadas en el ramo de Mecánica de Vuelo y el trabajo con el nuevo protocolo de comunicación, esto incluye desde el encendido de los equipos hasta la puesta en marcha de todo el software. Además de incluir la solución a problemas comunes que se vayan notando durante el desarrollo y pruebas de funcionamiento del simulador, los que no necesariamente deban ser técnicos, sino que también incluya solución a confusiones que surjan de no tener experiencia con el equipo.

Capítulo 4

Metodología de trabajo

4.1. Simulador

Para reconfigurar el simulador de acuerdo con las necesidades y las condiciones de diseño establecidas en primera instancia se realiza un levantamiento de antecedentes “en terreno” de la configuración actual y sus capacidades en términos de lo que hace ahora mismo el simulador y lo que indique la documentación de cada componente, que implica registrar en un documento sencillo toda esta información del hardware además de realizar copias de seguridad de todo el software de los discos duros lo que permita poder analizarlo en caso de que se presente algún problema que surja durante el desarrollo y que no existía antes, para esto se clonaran o se mantendrán guardados los actuales hasta que se necesite acceder a esta información.

El nuevo hardware disponible en el laboratorio para el uso en el simulador también se analizará con su documentación para comparar sus capacidades con el hardware actual, esto permitirá empezar a diseñar la nueva configuración enfocada en simplificar la disposición y aumentar la compatibilidad con el software de interés.

La parte más crítica de la configuración de hardware es la tarjeta de video o GPU porque es esta la que define la cantidad de salidas de video disponibles y en un simulador se tienen múltiples pantallas que amplían el rango de visión horizontal. Se van a evaluar las alternativas existentes para poder maximizar la cantidad de pantallas por cada computador, reduciendo la cantidad de computadores ojalá a solo uno.

Para el software se preparará una disposición de discos duros y particiones del sistema que permitan una rápida restauración en caso de cualquier problema, por lo que se definirán

espacios de almacenamiento con propósitos específicos que se mantengan separados, el proceso mismo se realizara utilizando herramientas de administración de discos basadas en Linux como GParted.

Configurar los controles personalizados existentes en el simulador para asegurar la máxima compatibilidad con el software de simulación requerirá investigación de los protocolos que comúnmente se utilizan para interactuar con este hardware y los estándares que siguen, los que se van a implementar de acuerdo con las pautas disponibles.

4.2. Protocolo de comunicación

El comienzo del desarrollo del protocolo de comunicación consistirá en lograr dos cosas:

1. Comunicación entre el simulador de vuelo a nivel de sistema operativo y el hardware externo

Esto requiere principalmente del análisis de las capacidades de comunicación del hardware externo puesto que debido a su arquitectura especializada serán el factor limitante y se debe trabajar alrededor de estas, un lenguaje en común que compartan placas como Arduino y Raspberry es ideal entonces se analizaran soluciones como Micro Python o Circuit Python.

Se escribirán códigos prototipo que cumplan con la comunicación y se realizarán pruebas comparando velocidad y versatilidad a partir de las funciones disponibles estudiando la documentación o leyendo experiencias que compartan en internet otros desarrolladores, una función importante puede ser el tener la capacidad de correr modelos tipo “machine learning” con TinyML.

2. Preparar con librerías existentes la extracción y manipulación de variables del simulador

Ya existen soluciones que facilitan el acceso a las variables internas de una simulación en curso, se van a evaluar estas soluciones respecto a los objetivos de tener la mayor compatibilidad con software de simulación además de X-Plane 9. Se escribirá un código prototipo que extraiga y manipule estas variables internas de la simulación haciendo uso de las librerías en cuestión con tal de poder luego hacer una interfaz de este programa con el prototipo de comunicación a hardware externo.

La combinación de estos dos códigos resultara en la primera cruda comunicación entre la simulación y hardware externo, a continuación, se debe crear un protocolo que tenga en cuenta el tipo de información que se está transmitiendo y las acciones que el hardware externo quisiera

tomar en respuesta, esto se realizara creando una lista de condiciones usuales de trabajo y diseñando el protocolo bajo criterios de buscar las mínimas latencia y uso de ancho de banda posible pero manteniéndolo adaptable. Finalmente, este protocolo se implementará en lo que serán iteraciones de código final el que sea compatible con los estándares de los simuladores de vuelo y el hardware externo de interés.

3. Documentación

Con el simulador reconfigurado, se invitará a miembros de la carrera interesados a probar y trabajar con el equipo para obtener realimentación sobre su uso y verificar el cumplimiento de los objetivos que se buscaban, con posibilidad de realizar modificaciones menores en respuesta a lo que se vaya reportando, además de registrar problemas comunes que deban ser mencionados en la documentación del uso del simulador.

Finalmente, en base a la experiencia y los registros tomados durante el desarrollo se crearán manuales que describan el uso y procedimientos para realizar trabajos de interés con el protocolo de comunicación, que estarán disponibles de manera física y online en un repositorio dedicado al simulador.