### 

75.61 - Taller de Programación III

***FresaGL***

***Documento de arquitectura***

### **Alumnos:**

|  |  |  |
| --- | --- | --- |
| Nombre | Padrón | E-mail |
| Sorrentino, Luciano | 88976 | lsorrentino@fi.uba.ar |
| Obon, Daniel | 92272 | obi.arts@gmail.com |

# 

# 

[**Introducción**](#_1hm42473h8ne) **2**

[**Análisis de la problemática**](#_jpl20iy9gft7) **2**

[Relevamiento](#_ckd4txkw600b) 3

[Aplicaciones de terceros](#_dzo815wn4wpe) 5

[Software de control](#_dzo815wn4wpe) 5

[Mach3 D1.90.075](#_dzo815wn4wpe) 5

[Conversor CAD/CAM:](#_dzo815wn4wpe) 5

[RhinoCAM 2.0.2.10](#_dzo815wn4wpe) 5

[Herramienta de modelado 3D](#_dzo815wn4wpe) 5

[Rhinoceros 4 SR Service Release 9](#_dzo815wn4wpe) 5

[Plugin para Joystick](#_dzo815wn4wpe) 5

[Xbox 360 Controller version 2.4.1](#_dqnimorhzpmq) 5

[Diagrama del sistema de control (lazo abierto)](#_oxkp1qlij16q) 5

[**Análisis de causales de falta de precisión en CNC**](#_syf9muyoh1iw) **6**

[Vibraciones y ruido inducido en los componentes](#_gis95t7s77d) 6

[Error inducido por la mecánica del sistema](#_p4nsk0bn95pi) 7

[Error inducido por PC e interfaz paralela](#_hhgpl4czb8f9) 7

[Demoras del Distribuidor al comandar los drivers](#_zbzdbcx1d1ym) 8

[Demoras del Driver al comandar al motor](#_it7ttgjkd4tf) 8

[**Solución propuesta**](#_jrftbp1v4i8h) **9**

[Consideraciones y alternativas de solución](#_6oztuopn79cz) 10

[**Arquitectura del sistema**](#_4dytvjd2smsp) **10**

[**Conclusiones,**](#_3k431eb87qwn) **20**

[**Perspectivas y trabajo a futuro**](#_n18rloykndvv) **20**

[**ANEXO**](#_2bghgwf87p9z) **22**

[Tabla de Mediciones](#_nx5tcimhsgit) 22

[Investigación de productos comerciales](#_ivvbyh2x0pw5) 23

# 

# Introducción

El canal de experiencias navales, ubicado en el subsuelo de la facultad, cuenta con una máquina fresadora que se utiliza para la fabricación de moldes de barcas. Se desea terminar de automatizarla de manera que pueda ser operada como una máquina CNC sin los defectos o imprecisiones que todavía se encuentran en el tallado.

Una fresadora CNC (Control Numérico por Computadora) es una máquina cortadora controlada por computadora que permite cortar distintos tipos de material. Puede realizar diversos trabajos de carpintería, como tallado, taladrado y modelado del material. Los principales beneficios que ofrece son los de automatización y precisión.

A partir de un modelo tridimensional realizado por un programa CAD, se genera el código que es interpretado por un sistema de control. Éste a través de unos drivers, opera los distintos motores para realizar el tallado del prototipo. Desafortunadamente los conversores del modelo 3D a instrucciones, no arrojan un resultado perfecto para todo tipo de máquina y es el caso de ésta fresadora.

Se plantea desarrollar un sistema que facilite la visualización y modificación de dicho código, pudiendo así alterar el comportamiento de la máquina hasta lograr un resultado satisfactorio.

# Análisis de la problemática

Una máquina CNC típicamente produce un trabajo consistente y de alta calidad y mejora la productividad de la fábrica. A diferencia de una copiadora de plantilla, la fresadora CNC puede ser tan eficaz al crear un solo modelo, como realizar reproducciones idénticas.

Asímismo una fresadora CNC puede reducir la cantidad de material desperdiciado.

Un software CAM convierte un modelo de diseño CAD en un código llamado Código-G, que puede ser interpretado por una máquina CNC. Es decir que son instrucciones sencillas que pueden ser transformadas en pulsos recibidos por los motores.

Descripción del problema

El laboratorio cuenta con un sistema de automatización a través de una interfase PC- Driver, que bajo exigencia de trabajo, no otorga la precisión deseada al momento de crear el modelo.

Asímismo el proceso de conversión del modelo tridimensional al código que maneja el software de control, introduce operaciones no soportadas por la máquina. Ésto genera la necesidad de que un operario esté monitoreando constantemente el proceso en caso que dichas operaciones deformen el producto final. Este trabajo se realiza observando cómo talla la máquina en conjunto a las instrucciones que se ejecutan. El proceso resulta tedioso y laborioso para el operario que utiliza el sistema.

Estado actual del sistema

A continuación se hará mención del relevamiento que se realizó de la planta. Desde el aspecto de hardware como así el de software.

#### Relevamiento

|  |  |
| --- | --- |
| **PC de escritorio** | |
| Sistema Operativo | Microsoft Windows XP Professional |
| Version | 5.1.2600 Service Pack 2 |
| Fabricante | Microsoft Corp. |
| Nombre del Sistema | PC\_FRESA |
| Fabricante del Motherboard | Gigabyte Technology Co. Ltd. |
| Modelo de Motherboard | GA-MA74GM-S2 |
| Tipo de Sistema | X86-based PC |
| Procesador | X86-Family 16 Model 4 Stepping 2 AunthenticAMD 3013Mhz (Dual Core) |
| BIOS Version/Fecha | Award Software International, Inc. FB 03/03/2009 |
| Version del SMBIOS | 2.4 |
| Dispositivo de inicio | /Device/HarddiskVolume1 |
| Capa de abstracción de hardware | Version = &ldquo;5.1.2600.2180 (xpsp\_sp2\_rtm.040803-2158)&rdquo; |
| Memoria física total | 2048,00 MB |
| Memoria física disponible | 1,49 GB |
| Tarjeta de video | 1 x DVI-D port (Note 2) |

|  |  |
| --- | --- |
| **Interfase PC – Driver:** |  |
| Conector de interfase a PC | DB-25 |
| Conector de interfase a driver | PLT 6 Vías |
| Señales de control | por lazo de corriente |
| Señales de finales de carrera | por lazo de corriente |
| Corriente de lazo | >2.5mA |

|  |  |
| --- | --- |
| **Motor Paso a paso** |  |
| Modelo | 34HS9803 (NEMA 34) |
| Bobinado | Bipolar |
| Torque de mantenimiento | 4.2 N/m |
| Tensión de alimentación | 40-160 Vdc |
| Corriente media | 2.7A (4A Máx) |
| Angulo de paso | 1.8° (200 PPV.) |

|  |  |
| --- | --- |
| **Driver de Motor** |  |
| Control Manual (HMI) | 2 pulsadores de comando (2 velocidades) |
| Control por interfase | Opto aislada 4.4 KV / 2.5mA / 10μs |
| Conector de Interfase | PLT 6 Vías |
| Conectores de alim. y motor | Bornera enchufable |
| Configuración de modo Half/Full | Por jumper de selección. |
| Señalización de fase de motor | 4 Leds de alta eficiencia. |
| Señalización de estado/dirección | 1 Led bicolor |
| Tensión de lógica de control | 8 -20 Vdc |
| Tensión alimentación de motor | 40 – 80 Vdc |
| Corriente máx. de fase de motor | < 4 A |
| Frecuencia máxima de pulsos | 10.000 pasos por segundo (PPS). |
| Micropasos | Full / Half Step |
| Dimensión | 150 x 88 x 25 mm. |

#### 

#### 

#### 

|  |  |
| --- | --- |
| **Aplicaciones de terceros** | |
| Software de control | Mach3 D1.90.075 |
| Conversor CAD/CAM: | RhinoCAM 2.0.2.10 |
| Herramienta de modelado 3D | Rhinoceros 4 SR Service Release 9 |
| Plugin para Joystick | Xbox 360 Controller version 2.4.1 |

#### 

#### Diagrama del sistema de control (lazo abierto)

El siguiente diagrama describe el sistema que controla la fresadora, transformándola así en una máquina CNC.

Los límites en los distintos ejes, conocidos también como finales de carrera, están conectados directamente a la placa que controla los drivers. Ésta contiene también un botón de parada de emergencia y se conecta con la PC a través de un puerto paralelo. Los drivers comandan los motores mandándoles pulsos respecto a lo que reciben de la interfase PC-Driver(Breakout Board).

El diagrama describe sólamente los ejes X e Y, pero en la planta existe también un eje Z análogo.

La fuente que alimenta al sistema está conectada a la misma fase que la PC, si bien tiene una llave térmica por seguridad, corresponde al mismo tablero y a la misma fase que todo el laboratorio.

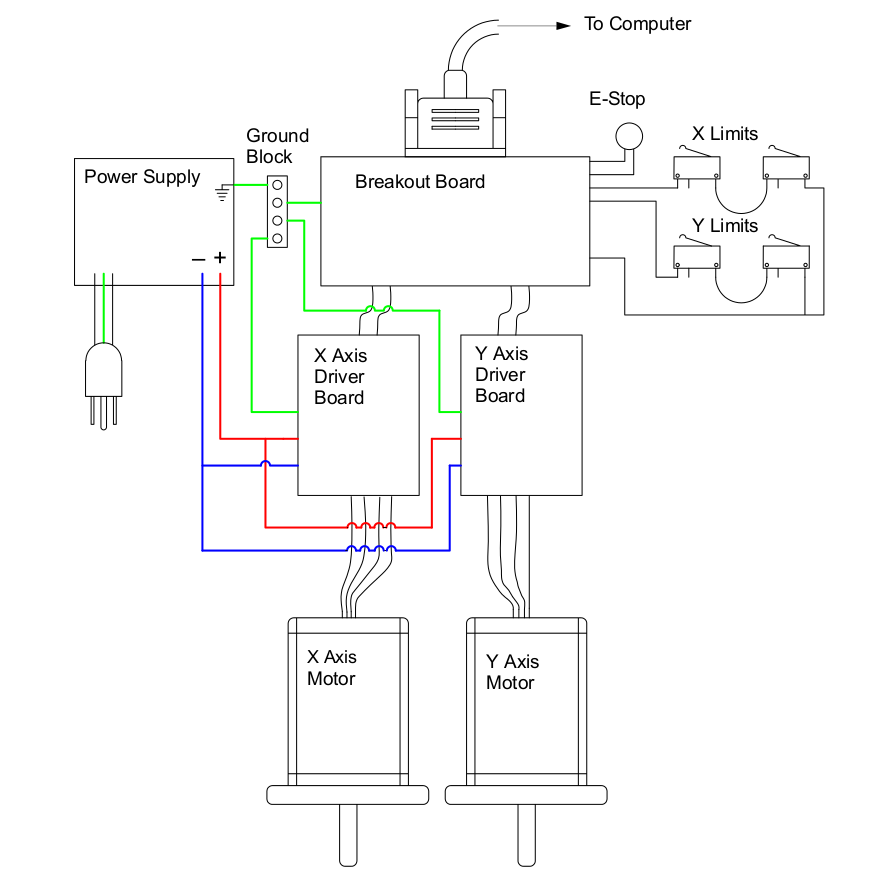


Imagen extraída del manual de instalación y configuración de Mach3. (Figure 4-12: Example Wiring Diagram)[[1]](#footnote-0)

# Análisis de causales de falta de precisión en CNC

En este apartado se analizan las distintas causas que se consideraron como posible fuente de imprecisión. Las mismas no pudieron verificarse debido que faltaba instrumentación o por estar fuera del alcance y presupuesto del proyecto. Surgieron a su vez varios inconvenientes que dificultaron o impidieron algunas experiencias[[2]](#footnote-1). No se permitieron hacer mediciones con las cuchillas funcionando por motivos de seguridad.

#### Vibraciones y ruido inducido en los componentes

Las cuchillas que cortan el material están montadas sobre un motor que funciona de manera contínua. Al igual que los electrodomésticos, estos motores eléctricos al girar inducen campo electromagnético. Por lo tanto, al conectarlos y más aún al forzarlos a girar cortando el material, generan ruido eléctrico[[3]](#footnote-2) . Teniendo en cuenta que el motor de una batidora a 30 cm produce un campo eléctrico de 100 V/m[[4]](#footnote-3), se estimó que podría haber un efecto de inducción.

Por causas de seguridad, no se pudo realizar las experiencias necesarias para comprobar si este escenario era cierto, sin embargo es sugerible mantener las cuchillas bien afiladas y aislar lo máximo posible los motores de las cuchillas como sugiere el artículo *“Acoplamiento inductivo y cómo minimizar sus efectos en la industria“[[5]](#footnote-4)* de César Cassiolato

#### Error inducido por la mecánica del sistema

Al ser un artefacto mecánico, la fresadora puede cometer errores de tallado debido al juego de las piezas al aplicarles un torque. El tornillo sin fin del eje Y puede ser el más susceptible a este tipo de imprecisión, en contraposición, al eje X que contiene una caja reductora y al eje Z que se comunica a través de correas. Asímismo se relevó una falta de lubricación en algunos sectores del tornillo del eje Z.

Esta causal no fue profundamente analizada por la diferencia de competencia o aptitud con los integrantes del laboratorio.

Los integrantes del laboratorio constataron también que cuando se opera la máquina en el modo manual, los errores mecánicos se van corrigiendo a medida que el operario los va detectando.

#### Error inducido por PC e interfaz paralela

A priori no hay ningún indicador que señale que este canal de comunicación no tiene pérdidas. Si bien los fabricantes de computadoras personales reemplazaron el uso del puerto paralelo por los USB, algunos hobbistas electrónicos prefieren utilizar un puerto paralelo para la comunicación entre sus placas y las computadoras. Es muy frecuente hallar esta tecnología cuando se trabaja con CNC.

Dependiendo del sistema operativo, la manera de despachar información a través del puerto paralelo, puede variar de manera directa (LINUX)[[6]](#footnote-5) o de manera acumulada o *“buffereada”* (Windows)[[7]](#footnote-6). Esto permite prever que el software relevado no puede actuar de manera inmediata y automática al obtener datos externos.

La consideración de esta causal se vió reducida al momento de hacer mediciones de los movimientos de la fresadora sin conectar los motores de las cuchillas. Los resultados indicaron que la falta de precisión no se producen por una pérdida en esta comunicación. Esto se aprecia porque no se indujeron errores en las corridas realizadas.

En la sección de Anexo se adjunta una tabla descriptiva de las mediciones

#### Demoras del Distribuidor al comandar los drivers

Esto puede ocurrir cuando el hardware de la interfaz entre la PC y los drivers no tengan la capacidad de computación necesaria para procesar los resultados de los drivers en conjunto con las instrucciones recibidas de la PC.

El objetivo del distribuidor es sincronizar los movimientos de los motores de manera de que cada instrucción recibida termine su ejecución en cada motor al mismo momento. A su vez recibe información de control del lado de la máquina (finales de carrera) para cortar la ejecución e informarle al software que se activó una parada de emergencia.

Se pueden observar pérdidas de pasos debido a que el hardware no logra transformar y despachar las microinstrucciones a los drivers con la velocidad emitida por el software. La consecuencia de esto es un desfasamiento del estado del driver, afectando el comportamiento del mismo al recibir nuevas microinstrucciones.[[8]](#footnote-7) Si el driver recibe una microinstrucción fuera de fase, simplemente la descarta.

Un posible escenario de evaluación para ver si se encuentran fallas en este punto crítico, es corroborar a través de un osciloscopio, si las señales emitidas por la breakout board corresponden a lo especificado. De no ser así o se presentase alguna incertidumbre, se podría reemplazar la placa distribuidora con un arduino, agregando si hace falta microcontroladores más potentes que el utilizado, y emular su comportamiento.

#### Demoras del Driver al comandar al motor

Los motores PaP a diferencia de los Servo, no necesitan de encoders que cuenten la cantidad de giros realizados debido a su gran precisión[[9]](#footnote-8). Los motores en el laboratorio siguen las reglas de estandarización donde se consideran 200 pasos por vuelta, lo cual indica que por pulso recibido se mueven 1.8°. En adición, las cajas reductoras conectadas a los mismos, reducen el error aún más. Debido a esto, se considera que si la causa se encontrara en esta interfaz, es debido a que no llegó el pulso al motor.

Como fue explicado en la contingencia anterior, se debe diferenciar entre un paso omitido y uno perdido. Se considera un paso perdido a la no detección del pulso entre el driver y el motor, debido a ruido eléctrico y paso omitido a la acción del driver de no enviar un pulso al motor por un descarte de microinstrucción. Los desfases pueden ocurrir por una mala configuración de la rampa de excitación de la señal cuadrada obtenida por el driver, donde se considera que el motor está detenido y se recibe una señal de “seguir moviéndose”.

Una posible solución a esta problemática es modificar la configuración de los pulsos, es decir, configurar la rampa y la altura de los pulsos que recibe cada eje. Modificar la cantidad de pulsos a mandar por unidad de tiempo, también podría ayudar a solucionar este efecto.

# Solución propuesta

Motivación

Realizando un análisis que refleja lo comentado en el apartado anterior, se pensó en una solución que resulte como paliativo a los problemas descritos.

Con el objetivo de caracterizar el sistema existente se realizó una comparación somera de la algunas soluciones comerciales existentes (ver anexo).

En cada uno de los casos se practicó un relevamiento de la documentación disponible, características técnicas y precios publicados al 7/12/16, que se expone en la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **KFlop y KStep**[[10]](#footnote-9) | **Sound logic encoder interface***[[11]](#footnote-10)* | **SINUMERIK 808D**[[12]](#footnote-11) |
| Proveedor | Dynomotion | Romaxx[[13]](#footnote-12) | Siemens |
| Cambios de Hardware | Breakout Board en caso de adquirir solamente KFLOP.  Drivers en caso de adquirir KSTEP. | Breakout Board, adición de encoders conectados al producto | PC, Breakout board y drivers |
| Cambios de Software | Plugin opcional para Mach3 | Plugin para Mach3, lo utiliza sólo como DRO[[14]](#footnote-13) | Software propietario embebido en el producto |
| Requerimientos de Instalación | Puerto paralelo funcionando, y conexión de pines con los motores y los drivers en caso de adquirir sólo KFLOP | Puerto paralelo funcionando, conexiones hacia los drivers y los encoders a traves de los pines | Sólo conexión con la red eléctrica |
| Capacidad de interacción programática | Interfaz a través de bibliotecas .NET | Producto cerrado, la interacción con el mismo es a través del plugin de Mach3 | Software propietario, SINUMERIK style programming language |
| Principales Beneficios | Capacidad de procesamiento de las señales, compatibilidad entre componentes comprobada | Control cerrado a nivel hardware, (Encoder - BreakoutBoard - Drivers) | Simpleza en un producto cerrado, confiabilidad en garantía por ser primera marca |
| Precio | U$D 1.045,00 | U$D 168,50 | U$D 11.990,50 |

##### Consideraciones y alternativas de solución

El análisis nos llevó a intentar atacar el problema en otro punto de la planta, debido a la imposibilidad del laboratorio de acceder a dichos productos.

Paralelamente, los productos comerciales sólo solucionaban el problema de la falta de precisión. Consideramos entonces plantear una alternativa que altere el sistema en un punto de abstracción más alto.

La solución planteada comprende la posibilidad de modificar el código generado por el conversor, como así visualizar gráficamente los movimientos que serán emulados por la fresadora. Editando el archivo con las instrucciones, el usuario va a poder modificar el comportamiento de la planta. Se le ofrece la posibilidad de cambiar la velocidad en la que opera la máquina en cierta porción de trabajo de tallado de una manera simple, como así también generar puntos intermedios entre los generados por el conversor. Estas variables modifican la respuesta de la planta a niveles de sucesivos. Al mandar una mayor cantidad de pulsos, los drivers se comportan de manera diferente, como así también al obligarlos a trabajar a otra velocidad.

Éste, como cualquier sistema CNC, el control sobre la planta debe ser a lazo abierto. Si se encontrase algún error, la máquina no puede repararlo debido a que el material ya fue tallado.

Teniendo en cuenta el alcance, presupuesto y tiempo para realizar el proyecto, se propuso esta alternativa económica.

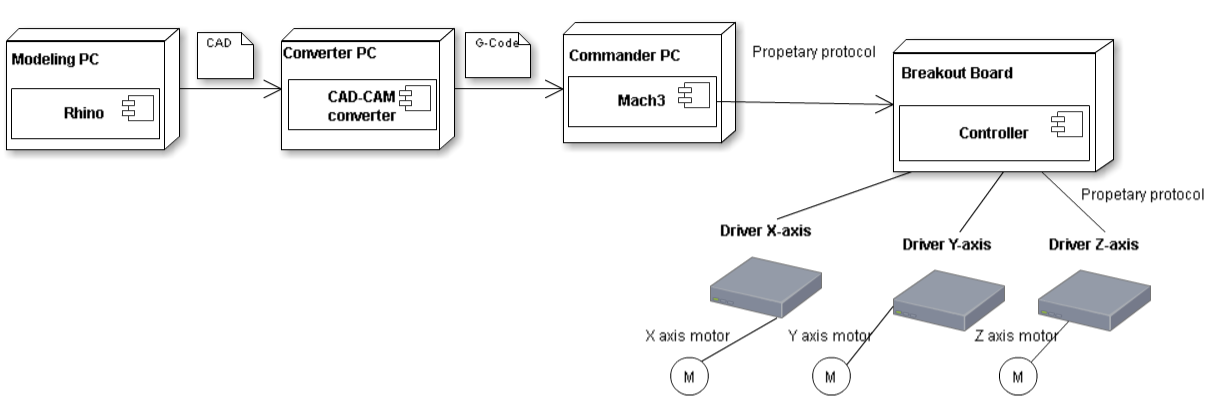
# Arquitectura del sistema

La arquitectura del sistema se explicará a través del modelo de cuatro vistas más uno.

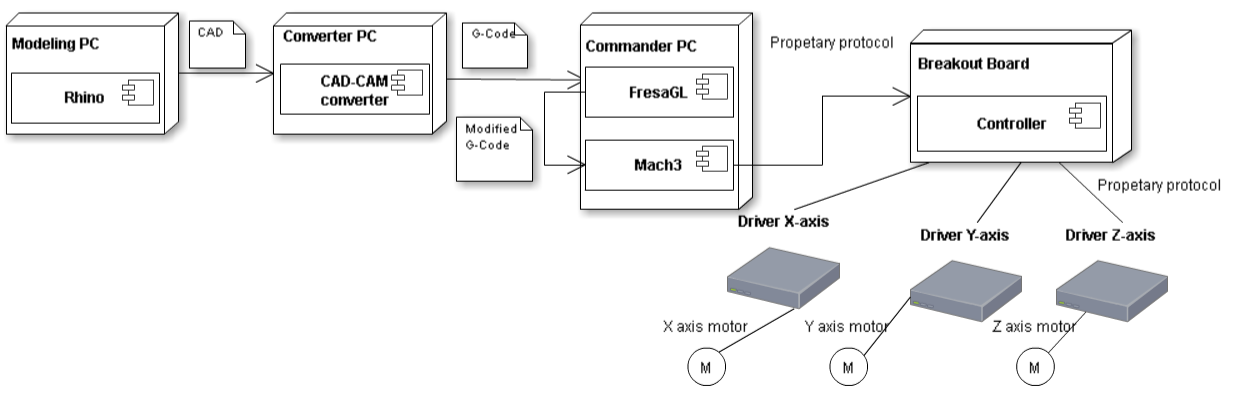
*Vista física*

La vista física describe al sistema desde el punto de vista de un ingeniero en sistemas, donde la se ilustran las conexiones entre los componentes. Es explicada a través de dos diagramas de despliegue: uno de cómo encontramos la planta y otro mostrando dónde actúa el sistema desarrollado.

Este diagrama grafica la planta que se halló al momento de comenzar el proyecto[[15]](#footnote-14). Se dejan ver las conexiones entre los distintos componentes.

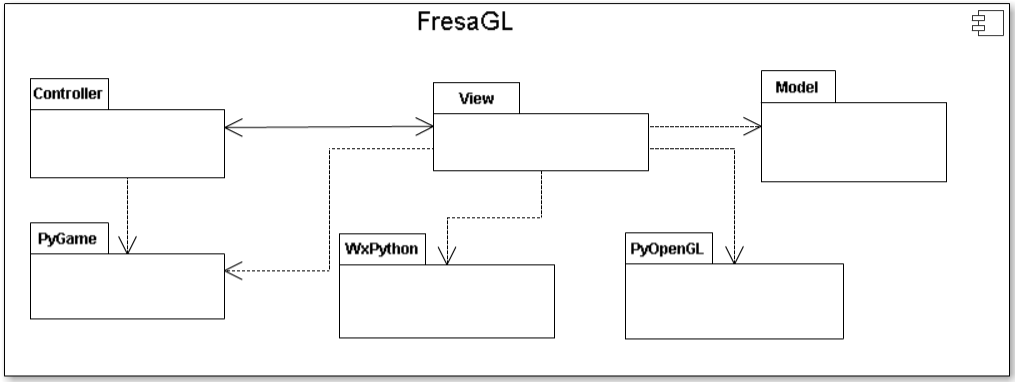


Nuestro sistema se ubica entre el conversor CAD/CAM y el Mach3 realizando un postprocesamiento del código G para detectar y corregir o al menos intentar mitigar lo errores producidos durante las primeras etapas del proceso.



*Vista de desarrollo*

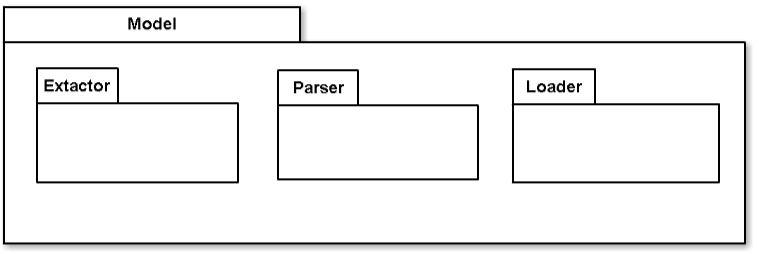
Un diagrama de paquetes, expone el sistema desde la visión del programador, separando así los distintos artefactos de software.



Nuestro sistema utiliza la arquitectura MVC para realizar separaciones lógicas de componentes de software:

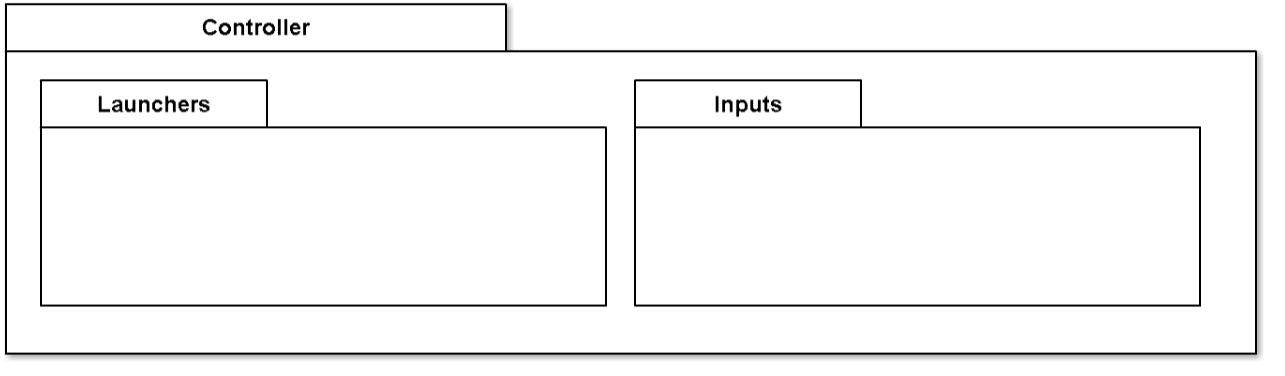
* Controlador: Ante una acción del usuario, indica al modelo la modificación que debe realizar y actualiza la vista para que refleje dicha modificación.
* Vistas: Permite la visualización del modelo tanto en forma de un gráfico 3D que permite visualizar los movimientos de la máquina, como en forma de una ventana de texto que permite ver el código G modificado, también permite comunicar al controlador las acciones del usuario a partir de una ventana de acción.
* Modelo: Permite la carga y posterior edición del archivo de código G.

Para el manejo del programa se utilizó la librería pyGame, para la renderización 3D se utilizó pyOpenGL y para las ventanas de visualización de código y acción se utilizó la librería WxPython.

El modelo hace uso de tres paquetes:

* Loader: Permite la carga y limpieza (pre formateo) del archivo de código G.
* Extractor: Usada para extraer los puntos por los que se moverá la máquina a partir del código G.
* Parser: Usados para realizar un “decorado” del código G, modificándolo según lo indique el usuario.

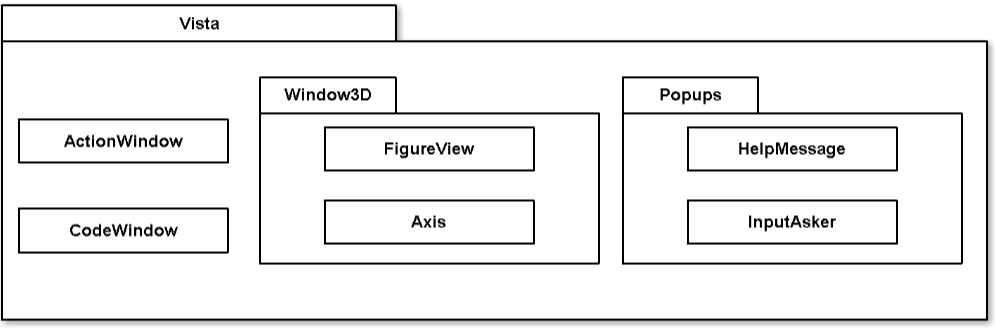
El controlador está dividido en varias clases que se encargan de manejar las distintas formas que tiene el usuario de comunicarse con nuestro programa:



* Un controlador principal, que es el que puede ver al modelo y las vistas y actualizarlos de ser necesario.
* Diversos controladores para manejar entrada por teclado, movimientos de cámara en la vista 3D, botones de la ventana de acción, cambios en el rango de selección, etc.
* También se tiene un paquete encargado del lanzamiento de programas externos, tales como el mach3 y el editor de texto.

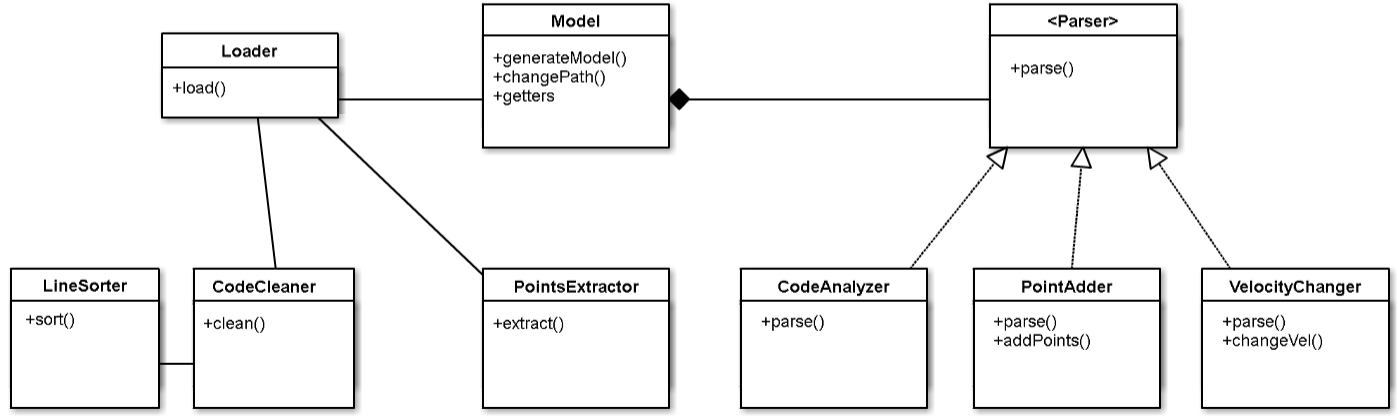
La vista está dividida en las tres ventanas principales del programa:

* Ventana 3D: Encargada de mostrar los movimientos que realizará la fresadora al momento de tallar el modelo descrito por el código G, contiene clases encargadas del renderizado del modelo y un eje 3D utilizado como referencia.
* Ventana de acción: Es la ventana con la que el usuario interacciona con el sistema, los cuáles se comunican con los controladores para ejecutar las modificaciones al modelo y la vista que se requieran.
* Ventana de código: Es la ventana encargada de mostrar al usuario el código G sobre el que se está trabajando.
* También hay un paquete que crea los diversos cuadros de diálogo usados para pedir información al usuario.



*Vista lógica*

Se eligió realizar un diagrama de clases del modelo por ser la parte más representativa de nuestro sistema.



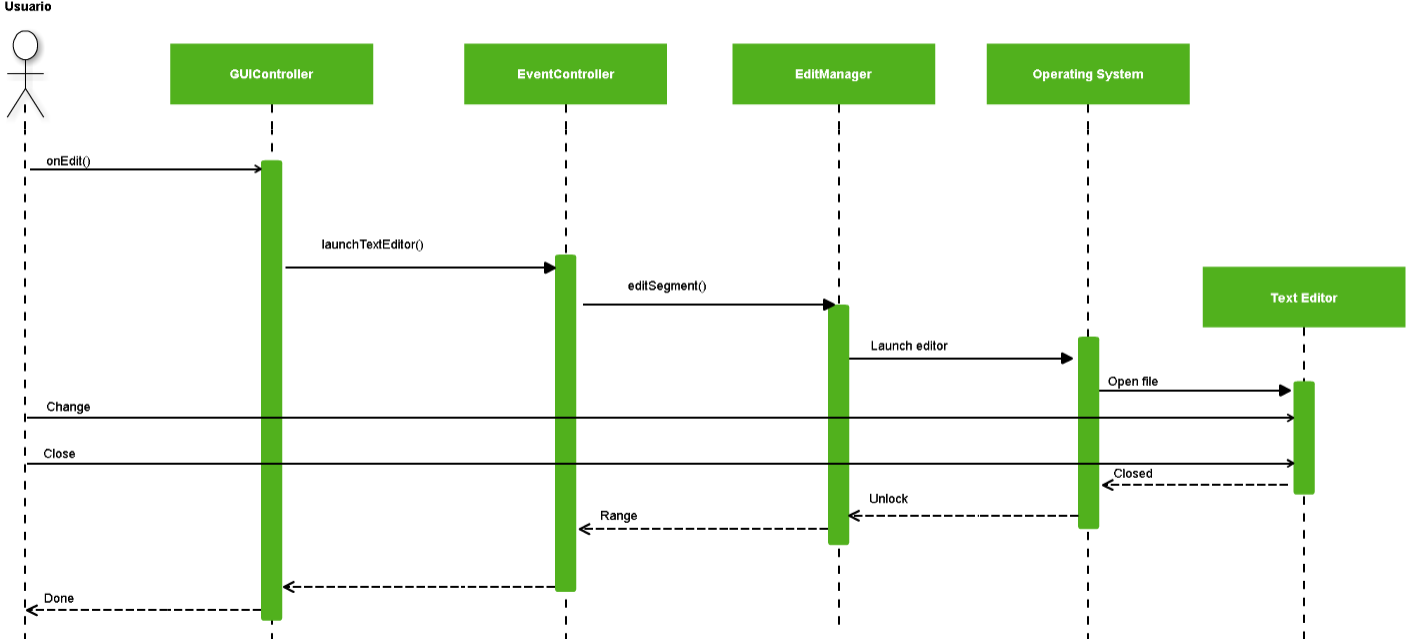
El modelo hace uso de una clase Loader que realiza la carga y una limpieza del código G y lo prepara para el posterior procesamiento, el cual es realizado por una serie de parsers que se ejecutan secuencialmente para “decorar” el código G según lo requiera el usuario.

*Vista de proceso*

Esta vista explica los procesos de sistema, intenta exponer al sistema desde su comportamiento en tiempo de ejecución. En este caso será representada por un diagrama de secuencia que expone cómo se lanza el editor desde nuestro sistema. A su vez se presenta otro de cómo se realiza la modificación automática del archivo al agregar puntos intermedios y los cambios de velocidad.

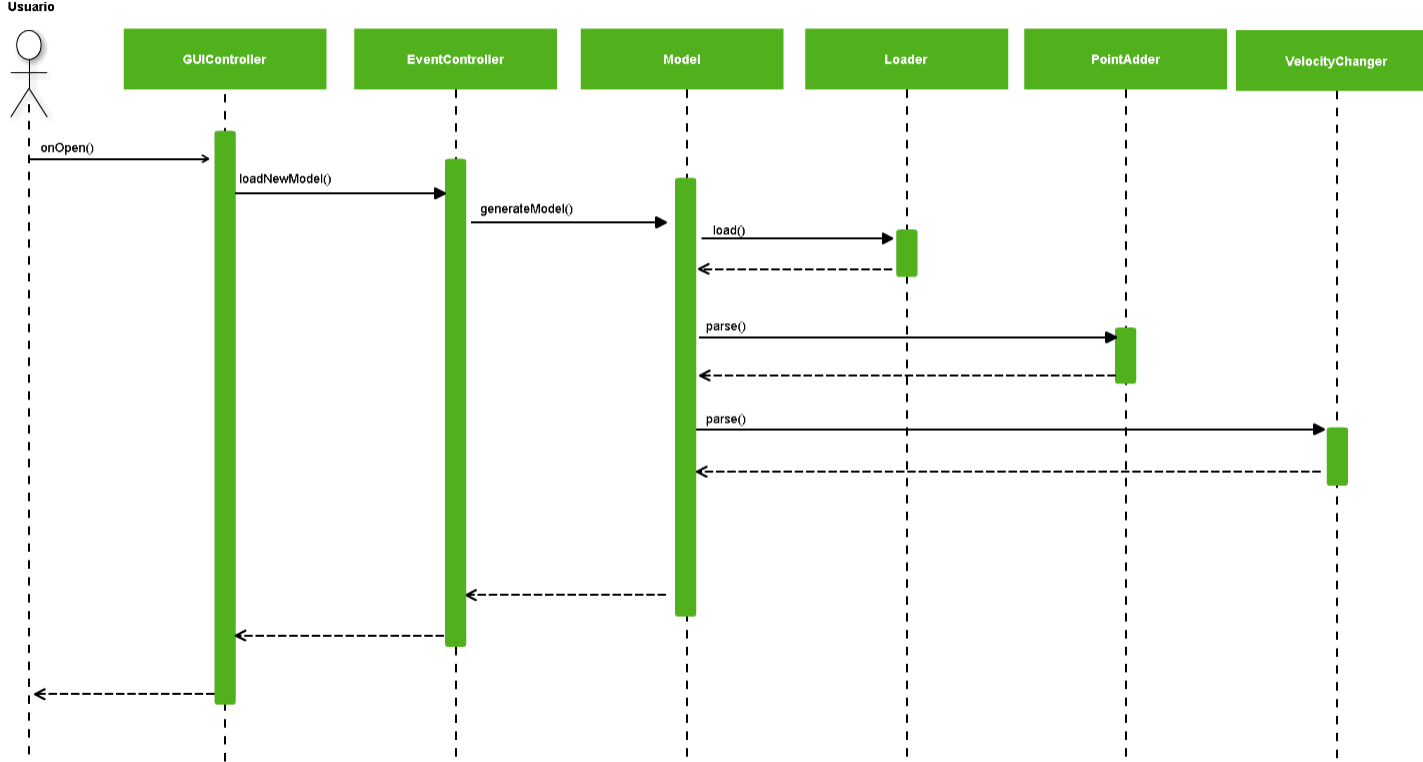
Edición manual de código:

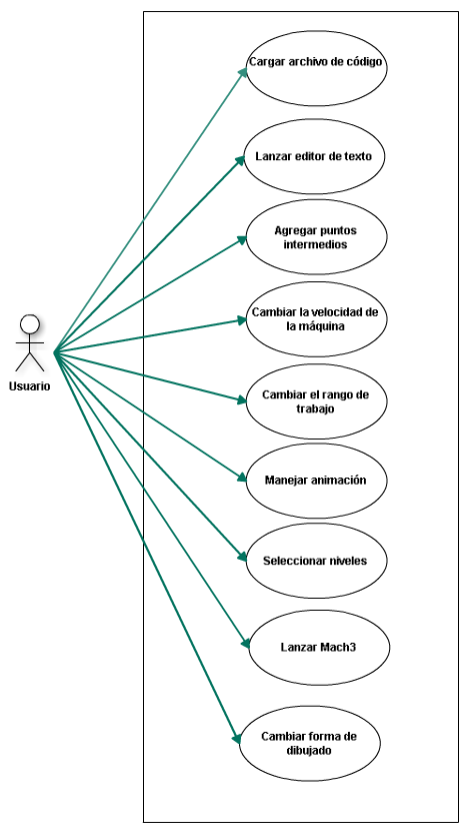
* El usuario hace click en el botón “lanzar editor de texto”.
* El controlador de la ventana de acción detecta que el usuario desea modificar manualmente la sección de código seleccionada y notifica al controlador principal que tiene acceso al modelo y la vista.
* Este oculta las ventanas del programa y llama al EditManager que se encarga de fraccionar el código par luego lanzar el editor de texto con únicamente la porción de código requerida.
* Se realiza una llamada al sistema para abrir el editor de texto con el código previamente mencionado.
* Una vez que el sistema operativo le retorna el control al programa, el EditManager retorna al controlador principal el código mergeado.
* El controlador principal vuelve a mostrar las ventanas ocultas previamente, y actualiza el modelo y la vista para reflejar los cambios al código G
* Se retorna el control al usuario.



Carga y modificación de Código G:

* El usuario hace click en el botón “cargar archivo” y selecciona el path del archivo a cargar.
* El controlador de la ventana de acción detecta notifica al controlador principal, el cual notifica al modelo.
* El modelo llama al Loader que realiza la carga y una limpieza del código.
* Luego el modelo aplica los distintos parsers para decorar el código y retorna el mismo modificado.
* El controlador principal actualiza la vista y se retorna el control al usuario.



Escenario de casos de uso

Cargar archivo de código:

|  |  |
| --- | --- |
| **Caso de uso** | Cargar archivo de código |
| **Descripción** | Carga del sistema operativo un archivo de código G |
| **Precondiciones** | Existencia del archivo en el file system |
| **Post-condiciones** | Archivo cargado en el sistema |
| **Flujo principal** | Seleccionar el archivo a abrir |

Lanzar editor de texto:

|  |  |
| --- | --- |
| **Caso de uso** | Lanzar editor de texto |
| **Descripción** | Envía al sistema operativo la orden de abrir un editor de texto con la porción de código G seleccionada |
| **Precondiciones** | Selección de rango de trabajo |
| **Postcondiciones** | Editor de texto abierto con el código cargado |
| **Flujo principal** | Divide el archivo de código, obteniendo la parte seleccionada, genera un archivo y lanza el editor de texto con ese archivo abierto. Finalmente reensambla las distintas partes |

Agregar puntos intermedios:

|  |  |
| --- | --- |
| **Caso de uso** | Agregar puntos intermedios |
| **Descripción** | Modifica el rango de código seleccionado, agregando puntos intermedios según si la distancia entre puntos supera o no una tolerancia |
| **Precondiciones** | Tolerancia positiva y código cargado |
| **Postcondiciones** | Archivo output con puntos intermedios adicionados |
| **Flujo principal** | Se recorre línea a línea el código y se extraen los puntos a los que corresponden dichas líneas. Si la distancia entre dos puntos consecutivos supera la tolerancia, se agregan puntos intermedios haciendo cumplir la condición |

Cambiar la velocidad de la máquina:

|  |  |
| --- | --- |
| **Caso de uso** | Cambiar la velocidad de la máquina |
| **Descripción** | Modifica el rango de código seleccionado, de manera tal que asegura que dicho código se ejecute a la velocidad elegida. |
| **Precondiciones** | Velocidad positiva y código cargado |
| **Postcondiciones** | Archivo output con velocidad modificada |
| **Flujo principal** | Se recorre el archivo buscando el comando que indica la velocidad en la que opera la máquina durante la porción de código seleccionada. Se ingresa un comando que regula la velocidad al principio de dicha selección, luego se eliminan todos los comandos que modifican la velocidad dentro de la selección y se recupera la velocidad encontrada anteriormente al final de la sección. |

Cambiar el rango de trabajo:

|  |  |
| --- | --- |
| **Caso de uso** | Cambiar el rango de trabajo |
| **Descripción** | Determina el área seleccionada de código |
| **Precondiciones** | Código cargado |
| **Postcondiciones** | Área de código seleccionada |
| **Flujo principal** | Se guarda el estado del modelo en un archivo temporal, se asigna el punto inicial y final del rango para que posteriormente se trabaje a partir del archivo temporal, se actualizan las distintas vistas |

ManejarAnimacion:

|  |  |
| --- | --- |
| **Caso de uso** | Manejar animación |
| **Descripción** | Se permite modificar la velocidad con la que se mostrarán los cuadros de animación, como así también retroceder y reiniciar la animación. También posibilita cambiar si la animación se realiza punto a punto, por capas o se desea ver todo el modelo |
| **Precondiciones** | Código cargado |
| **Postcondiciones** | Animación configurada de acuerdo a lo requerido por el usuario |
| **Flujo principal** | Se obtienen los valores y los controladores asociados a los mismos le indican a la vista cómo tiene que cambiar. |

Seleccionar niveles:

|  |  |
| --- | --- |
| **Caso de uso** | Seleccionar niveles |
| **Descripción** | Permite seleccionar las distintas zonas de código que se encuentran entre un cambio negativo del eje Z |
| **Precondiciones** | Código parseado separados por nivel |
| **Postcondiciones** | Nivel requerido seleccionado |
| **Flujo principal** | Se busca el punto correspondiente a la línea de código a partir de la cual se quiere seleccionar el nivel y se retorna el nivel requerido |

Lanzar Mach3

|  |  |
| --- | --- |
| **Caso de uso** | Lanzar Mach3 |
| **Descripción** | Solicita al sistema operativo la ejecución del programa Mach3 con el archivo de código modificado como parámetro |
| **Precondiciones** | Mach3 instalado y archivo de output generado |
| **Postcondiciones** | Mach3 lanzado con el archivo cargado |
| **Flujo principal** | Se guarda el código en el archivo output, se genera la macro y se ejecuta Mach3 a través de un llamado de sistema |

Cambiar forma de dibujado:

|  |  |
| --- | --- |
| **Caso de uso** | Cambiar forma de dibujado |
| **Descripción** | Permite la selección del modo de dibujo |
| **Precondiciones** | Código cargado |
| **Postcondiciones** | Cambio reflejado en vista 3D |
| **Flujo principal** | Se selecciona el modo y se le indica al renderizador que dibuje el modelo con dicha modalidad |

# Conclusiones,

En el presente trabajo se hizo un análisis de los distintos puntos en donde se pueden atacar los distintos problemas que pueden afectar a la planta. En el apartado “Arquitectura del Sistema” se puede apreciar en la vista física las distintas interfaces entre los componentes.

Las mediciones que se realizaron durante el desarrollo del proyecto, indicaron que nuestro sistema no induce error. Esto quiere decir, que la alteración de la planta a través de nuestro producto, no empeora la precisión de la máquina.

La fresadora sigue teniendo un control a lazo abierto, debido a que no es posible reparar un error una vez detectado. Ésto se debe a la física del escenario, de manera que si las cuchillas cortaron mal una sección de parafina, no hay marcha atrás, el proceso de tallado debe reiniciarse con un bloque nuevo, modificando la sección de código en la que se encontró el error para que la máquina trabaje a menor velocidad o con una mayor resolución de puntos por milímetros.

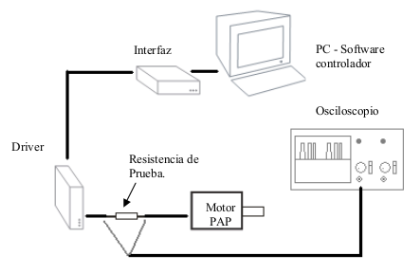
Sin embargo a continuación introducimos una serie de sugerencias a ser consideradas posteriormente.

# Perspectivas y trabajo a futuro

Se puede estudiar y modificar la manera que se comunican los componentes, como así reemplazarlos por otros. Al momento de hacer el trabajo se dejaron de lado algunas acciones aconsejables por falta de tiempo y alcance del proyecto.

Recomendaciones a futuro

* Realizar pruebas más extensivas con la máquina, incluyendo las cuchillas cortando y los tres ejes activos, para así detectar los errores antes mencionados y tener un mejor punto de partida al momento de ejecutar una nueva mejora.
* Analizar con un osciloscopio la señal que el break-out envía a drivers usando el benchmark recomendado por el tesista:



* Si lo anterior no arroja los resultados deseados, intercambiar el break-out board con un arduino o raspberry y probar si el sistema sufre algún impacto en cuanto a precisión.
* Considerar la posibilidad de implementar una nueva solución a nivel del conversor CAD/CAM y su configuración de forma que el archivo de código G sea mas sencillo de trabajar.

# 

# 

# ANEXO

## Tabla de Mediciones

Se generaron archivos de código G, describiendo formas sencillas tales cómo cuadrados, triángulos y círculos.Se ejecutaron dichos archivos con el Mach3 buscando medir los errores producidos, para luego, utilizando la versión beta de nuestro programa, ensayar algunas de nuestras soluciones soluciones propuestas para corroborar que el error medido se reduce, o al menos no aumenta.

Las herramientas de medición que se utilizaron, fueron las reglas métricas anexadas a la máquina.

|  |  |  |
| --- | --- | --- |
| Primera calibración | Tiempo | Error |
| Linea en x largo 2000 mm | 1:53 | sin error |
| Vuelta | 1:35 | sin error |
| Linea en z largo 500 mm | 0:38 | sin error |
| Vuelta | 0:39 | sin error |
|  |  |  |
|  |  |  |
| Línea en x largo 2000 mm con pasos de 200 mm | 1:43 | sin error |
| Vuelta | 1:42 | sin error |
| Línea en x largo 2000 mm con pasos de 100 mm | 1:50 | sin error |
| Vuelta | 1:44 | sin error |
| Línea en x largo 1000 mm con pasos de 100 mm | 1:50 | ½ mm |
| Línea en x largo 1000 mm con pasos de 5 mm | 1:55 | ½ mm |
| Línea en x largo 1000 mm con pasos de 2.5 mm | 2:40 | ½ mm |
| Línea en x largo 1000 mm con pasos de 1 mm | 3:51 | sin error |
| Línea en x largo 1000 mm con pasos de 1 mm para el otro lado | 3:58 | ½ mm |
| Línea en z largo 400 mm con pasos de 40 mm | 0:37 | sin error |
| Vuelta | 0:35 | sin error |
| Línea en z largo 400 mm con pasos de 20 mm | 0:40 | sin error |
| Vuelta | 0:41 | sin error |
| Línea en z largo 500 mm con pasos de 1 mm | 2:50 | sin error |
| Línea en z largo 500 mm con pasos de 1 mm para el otro lado | 3:01 | sin error |
|  |  |  |
|  |  |  |
| Triángulo en xz - largo X: 1000mm, largo Z:400mm | 2:13 | sin error |
| Triángulo en xz - largo X:1000mm, largo Z:400mm (30 puntos) | 2:20 | sin error |
| Cuadrado xz de 400mm en cada lado (volviendo al orig) | 1:45 | sin error |
| Cuadrado xz de 400mm en cada lado (volviendo al orig) con pasos de 40mm | 01:57 | sin error |
| Círculo en x-z de radio 200mm (termina en (x0-200,y0,z0)) 10 puntos | 01:27 | sin error |
| Círculo en x-z de radio 200mm (termina en (x0-200,y0,z0)) 30 puntos | 01:43 | sin error |
|  |  |  |
| Triángulo largo 3000 en x 500 en z | 05:30 | ½ mm en x |
| Triángulo largo 3000 en x 500 en z con puntos intermedios ( 14 puntos) | 05:44 | ½ mm en x |
| Triángulo largo 3000 en x 500 en z con puntos intermedios pasos de 1 mm | 27:13:00 | sin error |

No se logró observar un error significativo durante las experiencias, pudimos ver que las soluciones ensayadas no infieren error, pero hace falta repetir las mediciones con la fresadora cortando para saber si nuestras soluciones efectivamente reducen el error al momento de tallar.

## Investigación de productos comerciales

**KFlop y KStep de Dinomotion:**

A continuación se incluyen las características comerciales provistas por Dinomotion respecto del producto:

*“KFLOP es un novedoso controlador de movimiento que se encarga de los requisitos en tiempo real de su sistema desde el ordenador mediante el almacenamiento en un búfer de datos y la ejecución de varios hilos simultáneos de forma determinista. Combinado con la arquitectura de software que le da acceso fácil a funciones de bajo nivel y la capacidad de crear sus propias interfaces gráficas front-end para acceder a nuestras bibliotecas directamente a través de una interfaz .NET, el controlador de movimiento KFLOP es una poderosa solución a sus necesidades de control de movimiento.”*

*“KSTEP es una tarjeta de expansión de KFLOP diseñada específicamente para el control de motores paso a paso plug-and-play. Con sólo una alimentación, motores paso a paso y el KFLOP + KSTEP usted tiene todo lo que necesita para comenzar a controlar 4 motores”*

Con un precio de U$D 249 para KFLOP mas U$D 199 para el KSTEP, el precio final asciende a U$D 1.045

**‘Sound logic encoder interface’ de Romaxx:**

Reseña del producto según el proveedor:

*“4-axis encoder interface for use with Mach3 CNC control software. Closed encoder loop for precise positioning with 4 onboard relays and mutli-purpose inputs. For jog buttons, MPG's and more.”*

Este producto de Romaxx ofrece un sistema electrónico de control a lazo cerrado a nivel de comunicación breakout board - driver. Permite dejar a Mach3 con la única responsabilidad de actuar como DRO.

El precio del mismo es de U$D 168,5.

**SINUMERIK 808D de Siemens**

Descripción del producto según Siemens:

*“La solución compacta y amigable para el usuario que se utiliza para aplicaciones básicas de torneado y fresado.”*

Siemens ofrece un producto que automatiza completamente una máquina diseñada para trabajar con CNC. Ofrece una interfaz amigable con el usuario en forma de pantalla y botonera que le permite fácilmente al operario utilizar la máquina. Promete mantener aislado eléctricamente a sus circuitos y ofrece cursos de entrenamiento sobre el uso. Consta con un soporte para sus clientes y una garantía en caso de falla.

El precio expuesto en una tabla[[16]](#footnote-15) en su página web es de U$D 11.990

1. <http://www.machsupport.com/wp-content/uploads/2013/02/Mach3Mill_Install_Config.pdf> [↑](#footnote-ref-0)
2. Descrito en el manual de proyecto adjuntado a este documento [↑](#footnote-ref-1)
3. Consideramos ruido eléctrico, a todas aquellas señales de interferencias, de origen electromagnético, no deseadas. De manera que pueden alterar la señal a estudiar produciendo efectos que pueden ser perjudiciales. [↑](#footnote-ref-2)
4. http://www.who.int/peh-emf/about/WhatisEMF/es/index3.html [↑](#footnote-ref-3)
5. <http://www.smar.com/espanol/articulo-tecnico/acoplamiento-inductivo-y-como-minimizar-sus-efectos-en-la-industria> [↑](#footnote-ref-4)
6. Waugh, T. (1999). The Linux 2.4 Parallel Port Subsystem. *kernelbook. sourceforge. net/parportbook. pdf*. [↑](#footnote-ref-5)
7. <http://www.logix4u.net/parallel-port/16-inpout32dll-for-windows-982000ntxp> [↑](#footnote-ref-6)
8. https://www.machsupport.com/forum/index.php?topic=15204.10 [↑](#footnote-ref-7)
9. https://www.amci.com/industrial-automation-resources/plc-automation-tutorials/stepper-vs-servo/ [↑](#footnote-ref-8)
10. http://dynomotion.com/ [↑](#footnote-ref-9)
11. http://www.rogersmachine.net/encoderinterface.html [↑](#footnote-ref-10)
12. http://w3.siemens.com/mcms/mc-systems/en/automation-systems/cnc-sinumerik/sinumerik-controls/sinumerik-808/Pages/sinumerik-808.aspx [↑](#footnote-ref-11)
13. http://romaxxcncrouters.com/ [↑](#footnote-ref-12)
14. https://en.wikipedia.org/wiki/Digital\_read\_out [↑](#footnote-ref-13)
15. Trabajo Profesional – José Luis Casal – 83.338 Automatización de Fresadora de canal naval FIUBA por Control Numérico Computarizado. (2010) [↑](#footnote-ref-14)
16. https://w5.siemens.com/greece/internet/en/.../Simatic%20Price%20List%202016.xls  
     [↑](#footnote-ref-15)