UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA INDUSTRIAL
ESPECIALIDAD ELECTRÓNICA INDUSTRIAL

PROYECTO FINAL DE CARRERA

INTERFAZ GRÁFICA PARA LA MONITORIZACIÓN DEL GUIADO DE MÁQUINAS TUNELADORAS

AUTOR: JAVIER ROMERO CARRASCO
TUTOR: ALBERTO JARDÓN HUETE
COTUTOR: CARLOS GONZÁLEZ DE LA VEGA

Octubre de 2009

Son muchas las manos y los corazones
que contribuyen al éxito de una persona.

Walt Disney

El futuro pertenece a las personas
que creen en la belleza de sus sueños.

Eleanor Roosevelt

Agradecimientos

Quiero dar las gracias a mis padres, por mi oportunidad de existir, por su sacrificio en algún tiempo incomprendido, por su ejemplo de superación incansable, por su comprensión y confianza, por su amor y amistad incondicional, porque sin su apoyo no hubiera sido posible la culminación de mi carrera profesional. Por lo que ha sido y será… Gracias

Expresar mi más profundo agradecimiento a mi familia, porque gracias a su cariño, guía y apoyo he llegado a realizar uno de los anhelos más grandes de la vida, fruto del inmerso apoyo, amor y confianza que en mi se depositó y con los cuales he logrado terminar mis estudios profesionales que constituyen el legado más grande que pudiera recibir y por lo cual les viviré eternamente agradecido. Con cariño y respeto.

Gracias Nazareth, eres de esa clase de personas que todo lo comprenden y dan lo mejor de sí mismos sin esperar nada a cambio… porque sabes escuchar y brindar ayuda cuando es necesario…por estar siempre ahí de forma incondicional... por todo esto y mucho más, te quiero.

A todos mis amigos de Madrid y de Murcia, por su preocupación e interés, y por los grandes momentos que hemos vivido y disfrutado juntos.

A todos mis profesores y compañeros de la Universidad, que han hecho mucho más ameno el camino de mis estudios, por su amistad, compañerismo y respeto.

En especial, quiero dar las gracias a Rafael Portero, mi compañero de proyecto, por dar siempre lo mejor de sí mismo, ofrecer su ayuda incondicional y porque todos los momentos que he compartido con él, me han hecho crecer como persona.

Por último, dar especial agradecimiento a mi tutor Alberto Jardón, por la oportunidad que me ha brindado para realizar este proyecto y por la confianza que ha depositado en mí. Del mismo modo, agradecer a mi cotutor Carlos González todo el tiempo invertido en mí, por su paciencia e interés y por su apoyo moral en los momentos más difíciles.

A todo ellos, mi más sincero agradecimiento. Gracias.

Resumen

El presente Proyecto de Fin de Carrera, se centra en el desarrollo de una interfaz gráfica de guiado, la cual permite determinar y monitorizar en todo momento la posición y orientación de una tuneladora de hinca de tuberías durante la ejecución de una obra subterránea.

La interfaz gráfica se desarrolla según las especificaciones indicadas por los pilotos de este tipo de máquinas en la empresa, lo que supone replicar las interfaces comerciales utilizadas hasta el momento. Este interfaz se ejecuta en un PC y se ha desarrollado bajo entorno de programación MATLAB. Se comunica con un programa simulador de hinca de tuberías previamente implementado en LabVIEW y al software de captura de datos de una diana real. De esta forma, se monitorizan los datos de avance tanto simulados como reales.

El proyecto forma parte de los trabajos desarrollados por el Roboticslab bajo el Proyecto Nacional Científico-Tecnológico singular de carácter Estratégico "La Ciudad Multidimensional".

Índice General

Agradecimientos 5
Resumen 7
Índice General 9
Índice de Figuras 11

1. Introducción 15
1.1 Desarrollo de nuevas técnicas de guiado y control de las tuneladoras 16
1.1.1 Introducción 16
1.1.2 Objetivos 17
1.1.3 Ventajas de la construcción mediante máquinas tuneladoras 18
1.2 Objetivos y motivaciones del PFC 19
1.2.1 Objetivos 19
1.2.2 Motivaciones 20

2. Estado del Arte 22
2.1 Sistemas de guiado de máquinas tuneladoras 23
2.1.1 Hinca de tubos 24
2.1.2 Características del sistema de guiado de hinca de tubos 26
2.2 Análisis de las técnicas actuales y situación actual de la obra subterránea 27
2.3 Componentes de un sistema de guiado 29
2.4 Puesta en marcha y procedimiento básico del sistema de guiado 32
2.5 Parámetros a mostrar en un sistema de guiado 35
2.6 Posibles mejoras de los sistemas de guiado 42
2.6.1 Nuevos láser para la hinca de tubería con máquina tuneladora 42
2.6.2 Diseño de un prototipo para una nueva diana láser 43
2.7 Interfase Hombre-Máquina 45
2.8 Interfaces gráficas y entornos de programación gráfica 48
2.8.1 MATLAB (GUIDE) 50
2.8.2 LabVIEW 51
2.8.3 Delphi 52
2.8.4 Visual Basic 53

3. Desarrollo de la interfaz gráfica para el guiado 54
3.1 Especificaciones de diseño 55
3.1.1 Requisitos del sistema 56
3.1.2 Requisitos funcionales 59
3.1.3 Sistemas de referencia 59
3.2 Arquitectura del sistema de guiado 62
3.2.4 Procesado de información 64
3.3 Entorno de programación MATLAB 67
3.3.1 Entorno gráfico y creación de interfaces 67

3.3.2 Características 68
3.3.3 Entorno flexible de trabajo 69
3.3.4 Diseño y configuración de la GUI 74
3.3.5 Funcionamiento de una aplicación GUI 84
3.3.6 Flujo de operación 85
3.4 Implementación del software de la interfaz gráfica 86
3.4.1 Descripción del sistema 86
3.4.2 Programación del software 94
3.5 Comunicación entre los diferentes módulos software 121
3.5.1 Alternativas de comunicación entre LabVIEW y MATLAB 123
3.5.2 Alternativas de comunicación entre aplicaciones MATLAB 140

4. Resultados obtenidos 143
4.1 Manual de uso 144
4.1.1 Inicialización de la interfaz gráfica externa 144
4.1.2 Funcionamiento de la interfaz gráfica externa 146
4.2 Ejemplo de uso, resultados experimentales 155
4.3 Consejos y advertencias 162

5. Conclusiones 164
5.1 Conclusiones 165
5.2 Análisis crítico 167
5.3 Trabajos futuros y ampliaciones 169

6. Bibliografía 171

7. Anexos 174
ANEXO A. Hoja de características de la máquina tuneladora AVND 175
ANEXO B. Listado de las funciones del software 177

Índice de Figuras

Figura 1. Diagrama de hinca de tubería en emisario submarino..... 24
Figura 2. Componentes de la diana..... 29
Figura 3. Vista de la estación Leica TCA 1200..... 29
Figura 4. Vista de la plataforma donde se ubica la puntería anterior de referencia... 30
Figura 5. Vista de la cabina de control..... 30
Figura 6. Componentes del sistema de guiado..... 31
Figura 7. Vista de los componentes del sistema de guiado..... 33
Figura 8. Vista de la plataforma donde se ubica la puntería anterior de referencia 2 34
Figura 9. Vista de la pantalla principal del sistema de guiado PPS - TBM..... 35
Figura 10. Vista de la pantalla principal del sistema de guiado TBM..... 36
Figura 11. Vista de la pantalla principal del sistema de guiado SLS-T..... 37
Figura 12. Vista de la pantalla del sistema de guiado actual SLS-Microtunnelling.... 37
Figura 13. Desviaciones horizontales y verticales de la tuneladora..... 38
Figura 14. Rotaciones angulares de la tuneladora..... 39
Figura 15. Ángulo de incidencia. Vista en planta..... 39
Figura 16. Ángulo de inclinación. Secciones longitudinales..... 40
Figura 17. Ángulo de giro. Sección transversal..... 40
Figura 18. Vista del láser colocado en el pozo de ataque en una hinca de tubería 42
Figura 19. Vista del prototipo de láser..... 42
Figura 20. Vista del exterior del prototipo de diana 43
Figura 21. Vista del interior del prototipo de diana..... 43
Figura 22. Vista en perpendicular de la disposición de los componentes del prototipo de una diana con cámaras 44
Figura 23. Sistema HMI 45
Figura 24. Estructura general del software HMI..... 46
Figura 25. Comparación de Software para el desarrollo de HMI's..... 49
Figura 26. Interfaz gráfica en MATLAB 50
Figura 27. Interfaz gráfica en LabVIEW..... 51
Figura 28. Dimensiones reales de la máquina tuneladora..... 56
Figura 29. Diagrama de hinca de tubería en zona urbana..... 57
Figura 30. Visualización del sistema de guiado completo..... 58
Figura 30. Visualización del panel de control del simulador de hinca de tubos..... 58
Figura 32. Sistema de referencia global de la máquina tuneladora..... 60
Figura 33. Sistemas de referencia de la máquina tuneladora y de la diana 60
Figura 34. Puntos de cálculo de coordenadas de la parte delantera y posterior de la máquina tuneladora 61
Figura 35. Puntos de cálculo de la parte delantera y trasera de la flecha simbólica .. 61
Figura 36. Flujo de operación Simulador-Diana-GUI..... 63
Figura 37. Intercambio de variables entre los tres módulos que forman el sistema de guiado..... 66
Figura 38. Vista del escritorio de MATLAB (MATLAB Desktop) 70

Figura 39. Algunas páginas web sobre MATLAB.....74
Figura 40. Icono de inicio de la herramienta GUIDE.....74
Figura 41. Entorno de diseño: componentes etiquetados.....75
Figura 42. Configuración del color del fondo de pantalla.....76
Figura 43. Configuración del título y de los paneles.....77
Figura 44. Alineación de los componente de la interfaz gráfica.....77
Figura 45. Colocación de los textos estáticos.....78
Figura 46. Configuración del texto estático.....78
Figura 47. Botones de la interfaz gráfica.....79
Figura 48. Configuración de los botones de la interfaz gráfica.....79
Figura 49. Tabla de registros de guiado.....80
Figura 50. Configuración de la tabla de registros de guiado.....80
Figura 51. Interfaz gráfica de usuario.....81
Figura 52. Configuración del eje de coordenadas de la interfaz gráfica.....82
Figura 53. Configuración de los menús de la interfaz gráfica.....83
Figura 54. Presentación de la GUI en ejecución.....83
Figura 55. Ejecutar una GUI desde el directorio de trabajo.....84
Figura 56. Ciclo de operación de la GUI.....85
Figura 57. Diagrama de flujo de la función "presentacion.m" y "panel_total.m".....87
Figura 58. Diagrama de flujo de la función "representacion_2d.m".....90
*Figura 59. Nombre de celda donde se almacenan los diferentes tipos de parámetros*91
Figura 60. Diagrama de flujo de la función "representacion_3d.m".....92
Figura 61. Diagrama de flujo de las funciones de cálculo de datos reales y simulados.....93
Figura 62. Diagrama de flujo de las condiciones de representación 2D.....103
Figura 63. Dirección de la flecha para la función "grafica_1cuadrante.m".....105
Figura 64. Dirección de la flecha para la función "grafica_2cuadrante.m".....105
Figura 65. Dirección de la flecha para la función "grafica_3cuadrante.m".....105
Figura 66. Dirección de la flecha para la función "grafica_4cuadrante.m".....105
Figura 67. Diagrama de flujo de las condiciones de representación 3D.....107
Figura 68. Dimensiones y sistemas de referencia de la máquina tuneladora y la diana.....110
Figura 69. Funciones de cálculo reales y simuladas para cada punto de la flecha simbólica.....111
Figura 70. Dimensiones y sistemas de referencia de la flecha simbólica y la diana.111
Figura 71. Punto de referencia auxiliar de la diana.....112
Figura 72. Rotación "gamma" de la flecha simbólica sobre su eje longitudinal.....113
Figura 73. Distancia proyectada de la parte delantera o trasera de la flecha.....113
Figura 74. Rotación "alfa" y "beta" de la flecha sobre su eje vertical y trasversal....114
Figura 75. Diagrama del flujo del botón "Detener".....115
Figura 76. Diagrama de flujo del botón "Reanudar".....115
Figura 77. Diagrama de flujo del botón "Salir".....116
Figura 78. Diagrama de flujo del botón "Reiniciar".....116
Figura 79. Diagrama de flujo de la condición de salida de la GUI.....117

Figura 80. Condición de pausa y reanudación de la GUI 118
Figura 81. Diagrama de flujo del menú "grafica2d_real" 118
Figura 82. Diagrama de flujo del menú "grafica3d_real" 118
Figura 83. Diagrama de flujo del menú "grafica2d_simulada" 119
Figura 84. Diagrama de flujo del menú "grafica3d_simulada" 119
Figura 85. Diagrama de flujo del factor de escalado 119
Figura 86. Diagrama de comunicaciones del sistema de guiado 121
Figura 87. Comunicación entre los tres módulos y variables relacionadas mediante archivos de intercambio de variables 122
Figura 88. Gráfico de la composición de una librería dinámica 124
Figura 89. Selección del tipo de proyecto "Win32 Console Application" 125
Figura 90. Configuración de la aplicación a tipo "DLL" 126
Figura 91. Visualización del espacio de trabajo de Visual Studio 126
Figura 92. Declaración de las funciones de cabecera de la función "wrapper" 127
Figura 93. Declaración de las funciones de la función "wrapper" y declaración de "includes" 128
Figura 94. Estructura de la función "wrapper" de la librería dinámica 128
Figura 95. Añadir elementos existentes al proyecto 129
Figura 96. Visualización de los ficheros añadidos al proyecto 129
Figura 97. Seleccionar opciones del menú 130
Figura 98. Añadir directorios adicionales 130
Figura 99. Selección del nodo " MATLAB Script " 135
Figura 100. Seleccionar "Build Path" 135
Figura 101. Seleccionar "Paht Constant" 136
Figura 102. Seleccionar "Default Directory" 136
Figura 103. Selecciona categoría "Paths" y elegir la ruta para el "Default Directory" 137
Figura 104. Conexión de elementos para formar la ruta del nodo de "MATLAB Script" 137
Figura 105. Crear controles para las entradas añadidas 138
Figura 106. Crear indicadores para las salidas añadidas 138
Figura 107. Esquema final de las conexiones del nodo de "MATLAB Script" 139
Figura 108. Panel frontal de las conexiones del nodo de "MATLAB Script" 139
Figura 109. Interfaz gráfica de guiado en modo representación en dos dimensiones 145
Figura 110. Interfaz gráfica de guiado en modo representación en tres dimensiones 145
Figura 111. Paleta de herramientas de la interfaz gráfica externa 149
Figura 112. Menú contextual de la herramienta de "Rotación 3D" 150
Figura 113. Menú contextual de la herramienta "Cursor de datos" (1) 151
Figura 114. Menú contextual de la herramienta "Cursor de datos" (2) 151
Figura 115. Menú contextual de la herramienta "Vista Panorámica" 152
Figura 116. Opciones del menú "Representación Real" de la interfaz gráfica 152
Figura 117. Opciones del menú "Representación Simulada" de la interfaz gráfica .. 153

Figura 118. Opciones del menú "Ayuda" de la interfaz gráfica..... 153
Figura 119. Tabla de registro de guiado..... 154
Figura 120. Representación gráfica de la trayectoria rectilínea de la máquina tuneladora en dos dimensiones..... 155
Figura 121. Representación gráfica de una trayectoria rectilínea de la máquina tuneladora en tres dimensiones..... 155
Figura 122. Representación gráfica de una trayectoria de elevación en dos dimensiones..... 156
Figura 123. Representación gráfica de una trayectoria de elevación en tres dimensiones..... 156
Figura 124. Representación gráfica de una trayectoria de giro horizontal en dos dimensiones..... 157
Figura 125. Representación gráfica de una trayectoria de giro horizontal en tres dimensiones..... 157
Figura 126. Representación gráfica de una trayectoria de giro horizontal y vertical en dos dimensiones..... 158
Figura 127. Representación gráfica de una trayectoria de giro horizontal y vertical en tres dimensiones..... 158
Figura 128. Representación gráfica de una trayectoria de giro sobre el eje longitudinal en dos dimensiones..... 159
Figura 129. Representación gráfica de una trayectoria de giro sobre el eje longitudinal en tres dimensiones..... 159
Figura 130. Representación gráfica de una trayectoria desplazada en dos dimensiones..... 160
Figura 131. Representación gráfica de una trayectoria desplazada en tres dimensiones..... 160
Figura 132. Carpetas donde se almacenan los registros de guiado reales y simulados..... 161
*Figura 133. Registros de guiado reales *.xls*..... 161
*Figura 134. Registros de guiado simulados *.xls*..... 161

1. Introducción

Este Proyecto de Fin de Carrera forma parte del Proyecto Científico-Tecnológico singular de carácter Estratégico "La Ciudad Multidimensional" financiado en su mayoría por el Ministerio de Educación y Ciencia.

Dentro de este proyecto, se encuentra la tarea de "Desarrollo de nuevas técnicas de guiado y control de las tuneladoras", la cual ha sido investigada y desarrollada por la Universidad Carlos III de Madrid conjuntamente con la colaboración de la empresa EUROHINCA, para la especialización y mejora del sistema de guiado de las máquinas tuneladoras.

En este capítulo se introduce al lector en el marco del desarrollo de nuevas técnicas de guiado y control de las máquinas tuneladoras, dando una visión global de las razones por las cuales surge esta investigación y de los objetivos que persigue. A continuación, se expondrán los objetivos y las motivaciones que persigue el presente Proyecto de Fin de Carrera.

1.1 Desarrollo de nuevas técnicas de guiado y control de las tuneladoras

1.1.1 Introducción

El proceso de tunelación es uno de los procesos constructivos más complejos y para el cual se necesita un alto nivel de seguridad. En este proceso, los temas de seguridad y precisión priman sobre la productividad. La prioridad de esta investigación se centra en el desarrollo de sistemas de tunelación bajo el concepto de seguridad total, que incluye un alto nivel de predicción de sus efectos sobre el entorno y el medioambiente.

Las actuales tuneladoras de excavación de túneles, son máquinas muy costosas y complejas que cuentan con un alto grado de sofisticación tecnológica. Sin embargo, es un hecho comprobado que su guiado no está optimizado.

La problemática actual que existe en torno a las tuneladoras se centra en dos aspectos fundamentales.

En primer lugar, es un hecho comprobado, que por lo general los operarios no son capaces de saturar las capacidades de las máquinas tuneladoras, fundamentalmente por un déficit de formación de los operadores y de los encargados del mantenimiento. La consecuencia de todo ello es que las máquinas son infrautilizadas debido a que no existen simuladores o interfaces réplicas de las reales que permitan un contacto previo a un futuro operario de tuneladoras para su aprendizaje.

En segundo lugar, los sistemas de guiado y control utilizados para referenciar las máquinas tuneladoras dentro del túnel, tienen un margen de error debido a la utilización de láseres rojos de mucha longitud de onda, ya que son afectados por las condiciones adversas del interior del túnel y las largas distancias de los túneles a construir.

Por este motivo, surge la necesidad de mejorar el rendimiento y la eficacia de los sistemas de guiado de estas máquinas. Para ello, se han desarrollado algunas mejoras de los sistemas actuales, como la utilización de un láser verde de menor longitud de onda que permite mayor alcance y precisión, y una diana que en vez de utilizar células fotosensibles a la luz láser, se implementa mediante cámaras ópticas.

El presente Proyecto de Fin de Carrera está enmarcado dentro de la investigación del desarrollo de nuevas técnicas de guiado y también va contribuir a la mejora de éstas mediante el desarrollo de una interfaz Hombre-Máquina (HMI), réplica de una interfaz gráfica real que permita realizar tanto simulaciones como afrontar situaciones reales. Además, esta interfaz gráfica monitorizará tanto la información gráfica como paramétrica de una manera mejorada y más fácil e intuitiva para facilitar la labor de aprendizaje y formación de los operarios.

Gracias a estas mejoras se pretende mejorar las condiciones de trabajo, precisión de guiado, autonomía y rendimiento del proceso de excavación de la obra subterránea. Además, al optimizar la eficacia de la tuneladora se provocaría una reducción de costes y tiempos de construcción significativa y aseguraría una alta productividad y una mayor garantía en los plazos de ejecución. Por último, gracias a la facilidad de la interfaz visual se asegura el aprendizaje del operario de una manera igual de instructiva, más rápida y con menos costes para la empresa.

1.1.2 Objetivos

Los principales objetivos se pueden resumir en cuatro ideas fundamentales:

- En primer lugar, el análisis de los sistemas actuales para la posterior investigación y desarrollo de nuevos sistemas o mejoras de los mismos. Se ha de realizar la construcción de un prototipo de simulador genérico que recoja, en la medida de lo posible, las características que se prevén para los simuladores del futuro, es decir, que sea configurable y con una interacción lo más directa con el sistema de proyección. Además debe ser simple y de fácil manejo la interfaz visual ya que se asegura el aprendizaje del operario de una manera igual de instructiva, más rápida y con menos costes para la empresa.

- En segundo lugar, creación de una interfaz gráfica de interacción máquina – terreno donde se puedan definir las bases que permitan la obtención de un módulo que sirva al controlador predictivo como base para obtener datos o parámetros necesarios, y así crear un sistema de guiado autónomo donde el usuario estaría sólo en modo supervisor.

- En tercer lugar, el sistema de guiado se basa en un láser, normalmente es de color rojo que no tiene alcance suficiente para la nuevas distancias planteadas. Por ese motivo, se plantea cambiar dicho láser rojo de 633nm. por uno de color verde de 532nm. que alcanza la diana sin problemas. Además el nuevo láser no presenta tantos problemas de refracción ante la suciedad del ambiente, muy importante para un correcto guiado.

- En cuarto lugar, construcción de un prototipo básico de diana para industrializarlo. Se han investigado el diseño de nuevos prototipos de dianas que sean sensibles al rayo láser verde. Así mismo, el estudio de la posibilidad de utilizar dianas que en vez de utilizar células fotosensibles a la luz láser, se implementaran mediante cámaras ópticas que puedan definir la posición del rayo dentro de la diana.

1.1.3 Ventajas de la construcción mediante máquinas tuneladoras

La construcción de un túnel con tuneladora tiene muchas ventajas tanto técnicas como medioambientales.

Frente el método de construcción tradicional de minería y zanja supone menores riesgos laborales y menor impacto ambiental, ya que se trabaja sin explosivos, no es preciso bajar los niveles freáticos, no hay contacto directo con el frente, no se producen desprendimientos y al mismo tiempo no se producen alteraciones en la superficie. Además supone una mayor producción y rapidez en los trabajos ya que reduce el movimiento de tierras y la climatología no afecta a la producción.

La utilización de tuneladoras para la ejecución de tuneles está siendo cada día más utilizada para la ejecución de proyectos como: colectores y redes de alcantarillado, cruces de carreteras, emisarios submarinos, desagües al mar de emisarios pluviales, desagües al mar de plantas depuradoras, tomas de agua salina para desaladoras, tomas de agua salina para piscifactorías, desagües de fondo para embalses, gaseoductos y oleoductos, conducciones eléctricas, paraguas de microtunel para cruces entre vías de comunicación, conducciones de tubería metálica a presión, etc.

1.2 Objetivos y motivaciones del PFC

1.2.1 Objetivos

El objetivo del presente Proyecto de Fin de Carrera es el desarrollo de una interfaz gráfica de guiado que permita monitorizar la posición y orientación de una hinca de tuberías durante la ejecución de una obra subterránea.

A continuación, se van a describir los objetivos específicos:

- Diseño de un panel de control para la interfaz gráfica externa lo más explícito e intuitivo posible para que el operario que dirija la máquina tuneladora se adapte con facilidad al control paramétrico y a la monitorización de la dirección del sistema de guiado.

- La interfaz en modo de monitorización de datos simulados debe servir como consola virtual para enseñar y formar previamente a los operarios antes de la conducción real de la máquina tuneladora. Este sistema debe ofrecer un alto nivel de realismo ya que la consola virtual es análoga a la interfaz gráfica real.

- Creación de varios menús implícitos en el panel de control de la interfaz, que permitan elegir al operario el modo de representación gráfica a realizar, entre dos o tres dimensiones, y el tipo de datos entrantes que desea recibir, entre reales o simulados.

- Registro tanto de los parámetros reales como de los simulados, generados por la máquina tuneladora, para su posterior análisis mediante la creación de un fichero de almacenamiento.

- Creación de una interfaz gráfica externa mediante la utilización del programa MATLAB. Ésta posee la herramienta GUIDE (Graphical Use Interface Development Environment) que permite crear interfaces gráficas y dispone de las características básicas de todos los programas visuales y de interacción hombre-máquina. Estas son las llamadas GUI (Guide User Interface).

- Permitir realizar cambios de posicionamiento del sistema de referencia global de la máquina tuneladora.

- Representación de la orientación de la máquina tuneladora mediante una flecha simbólica de la misma forma que en los actuales sistemas de guiado.

- Realizar la comunicación del software de la interfaz gráfica con el software de la diana láser y el software del simulador de hinca de tubos.

1.2.2 Motivaciones

Las actuales tuneladoras son un gran avance en el mundo de la construcción. Están en constante evolución y son el futuro por la gran cantidad de ventajas y beneficios que suponen tanto para el medio ambiente como para la seguridad y los riesgos laborales. Son máquinas muy costosas y complejas que cuentan con un alto grado de sofisticación tecnológica. Sin embargo, es un hecho comprobado que su guiado no está optimizado, por ello, surge la necesidad de desarrollar un nuevo sistema de seguimiento con un márgen de error menor, con unas referencias globales y un alto nivel de repetitividad. Estas circunstancias adversas hacen necesario el desarrollo de nuevos sistemas multisensoriales y de nuevas técnicas de control inteligente de guiado de las máquinas. Además, se une el problema de que no existen simuladores o interfaces réplicas de las reales que permitan mejorar las habilidades técnicas de un operario ante situaciones adversas o incluso permitan un contacto previo a un futuro operario de máquinas tuneladoras para su aprendizaje.

Todos estos inconvenientes mencionados anteriormente han repercutido en la creación del proyecto "Desarrollo de nuevas técnicas de guiado y control de las tuneladoras" anteriormente descrito en el apartado 1.1, que se encargue de mejorar la precisión y la eficacia de los sistemas de guiado actuales. Además este proyecto debe hacerse cargo del desarrollo de un software lo suficientemente sofisticado para poder recrear en una interfaz gráfica situaciones prácticas lo más realistas posibles del entorno de una máquina tuneladora real.

El presente Proyecto de Fin de Carrera, "Interfaz gráfica para la monitorización del guiado de máquinas tuneladoras", forma una parte muy importante del proyecto global debido a que la interfaz gráfica de guiado va ser el futuro panel de mandos de la máquina tuneladora donde el operario podrá dirigir todo el proceso de control y seguimiento de la trayectoria real y simulada de la misma. Esta interfaz gráfica externa tiene que ser lo más explícita y compresible posible para que el operario se adecue lo más rápido posible a ella y pueda mejorar el control de la máquina tuneladora para aumentar su rendimiento.

Si los resultados obtenidos están acordes con los resultados previstos, se espera un gran salto tecnológico y revolucionario en los sistemas de guiado y control de las máquinas tuneladoras que va a ocasionar una gran suma de ventajas y beneficios.

En primer lugar, dar especial importancia a la mejora de los riesgos laborales ya que se va aumentar rigurosamente la precisión y fiabilidad en el guiado de las máquinas tuneladoras, siguiendo de manera fiel la trayectoria ideal produciéndose menos desviaciones y con un error más reducido.

En segundo lugar, se va producir un menor impacto medioambiental ya que el tradicional método por zanja con todos los inconvenientes que conlleva como levantamiento de tierras, rotura de cañerías y alteración de la superficie va quedar en desuso.

Por último, los niveles de rendimiento y productividad van a incrementarse significativamente ya que la climatología no afecta a la producción y se origina mayor rapidez y empuje en los trabajos.

Todos estás mejoras, tanto del sistema de guiado y de la interfaz gráfica externa como todas las ventajas y beneficios, hacen a la máquina tuneladora una gran apuesta de futuro y desarrollo y son las principales y más importantes motivaciones que llevan a trabajar con ilusión y esmero en este Proyecto de Fin de Carrera.

2. Estado del Arte

A continuación se introducirá al lector en las diferentes disciplinas que abarca este proyecto, presentándole para cada una de ellas los conceptos básicos, generalidades, clasificaciones, diseños de interés y explicación detallada para cada apartado expuesto, con el fin de que la lectura del Proyecto de Fin de Carrera pueda realizarse sin dificultad, incluso para aquellos que nunca hayan tratado con los temas aquí abordados. Las disciplinas que se revisarán son las siguientes: sistemas de guiado, componentes, puesta en marcha y procedimiento, parámetros a mostrar, análisis de las técnicas actuales y situación actual de la obra subterránea.

2.1 Sistemas de guiado de máquinas tuneladoras

En la construcción de túneles se emplean dos procedimientos diferentes, hinca de tubo y construcción de túneles mediante recubrimiento por dovelas o revestimientos de otro tipo. La utilización de cada procedimiento depende de muchos factores, entre ellos, la finalidad del túnel que se requiera realizar, las circunstancias del terreno, etc.

En la hinca de tubería el recubrimiento está en movimiento durante la construcción del túnel. Esto implica que dentro del túnel no se tienen puntos de coordenadas fijas y se necesita de un sistema de guiado que pueda medir ese desplazamiento de coordenadas para determinar la posición de la máquina tuneladora. Para la hinca de tubos se ha desarrollado un sistema de guiado láser con diana inteligente montado en primer lugar en una consola mural o en una columna de medición en el pozo de arranque, que se pueda designar como estable. Es el sistema de guiado más universal y sencillo. Válido para todo tipo de diámetros, longitudes y radios de curvatura, para hincas de corta longitud, rectas y curvas.

El sistema de guiado para construcción de túneles con resvestimiento de dovelas se ha utilizado en muchos de los túneles construidos con tuneladora de España como en la ampliación del metro de Madrid, Metrosur, Guadarrama, etc.
Está estructurado en dos niveles. El primer nivel está concebido para el uso del piloto. En él se muestra la posición y tendencias de la tuneladora, tanto gráfica como numéricamente, así como los tipos de dovelas a instalar. El segundo nivel está diseñado para el uso de los topógrafos de la obra. Desde este nivel se puede modificar el eje de proyecto, parámetros de la tuneladora, tipo de anillo a instalar, coordenadas del láser, etc. Para los túneles con recubrimiento de dovelas se ha desarrollado el sistema de guiado láser con diana inteligente montado al revestimiento del túnel en un área que se pueda designar como estable.

A continuación, se va a hacer hincapié en el procedimiento de hinca de tubos ya que es el sistema utilizado para el desarrollo del sistema de guiado.

2.1.1 Hinca de tubos

La realización de túneles mediante el método de hincas de tubo, es una práctica habitual en todo el mundo. Éstas pueden ser tanto verticales para pasar por debajo de ríos, carreteras, túneles, como horizontales para evitar cimentaciones de edificios, seguir el curso de una calle, etc.

En este tipo de perforación, se utiliza el tubo como elemento definitivo del túnel y al mismo tiempo como elemento de empuje sobre la tuneladora.

El avance se realiza gracias al empuje efectuado por un conjunto de cilindros de empuje instalados en el pozo de ataque sobre el tubo de hinca, el cual ha sido fabricado siguiendo unas normas estrictas, para poder soportar grandes esfuerzos longitudinales y transversales sin sufrir ningún deterioro.

El tubo situado sobre el bastidor forma parte del túnel una vez concluida la hinca. Cada tubo lleva instalada una junta en la boquilla, que debe garantizar la estanqueidad así como de una "sufridera" en la cola, que absorbe las posibles irregularidades del tubo y que permite que este no sufra al unir dos tubos y empujar hormigón con hormigón.

En todas las hincas se construye un pozo de ataque con un muro de reacción que soportara las presiones de empuje de toda la tubería y un pozo de llegada donde se rescata la máquina. Para hincas de gran longitud, se instalan unos elementos entre los tubos que se denominan "Estaciones Intermedias", y que permiten distribuir los esfuerzos entre varios tramos. Véase figura 1.

Figura 1. Diagrama de hinca de tubería en emisario submarino

La referencia principal del sistema se obtiene de un rayo láser visible que se emite desde una estación láser teodolito.

El láser teodolito se encuentra montado en primer lugar en una consola mural o en una columna de medición en el pozo de arranque, que se pueda designar como estable.

El rayo láser transcurre por una distancia de aproximadamente 100-200 m, dependiendo de la intensidad del láser, de las condiciones atmosféricas en el tramo de tubería y de la porción de refracción, hacia una diana, la cual se encuentra fijamente instalada en la parte frontal de la máquina tuneladora. La distancia útil entre el láser teodolito y la diana dependerá del tamaño de la ventana para el láser y de las condiciones de curvatura del tramo de tubería.

Una vez que el rayo láser hace blanco en la diana, se mide la ubicación exacta del punto de incidencia relativo al centro de la diana. También se determina el ángulo de incidencia horizontal del rayo láser en la placa frontal de la diana. En la diana se encuentra también integrado un inclinómetro biaxial para la medición de la inclinación longitudinal y de la rodadura de la diana.

La distancia entre la estación láser acompañante y la diana es algo variable, se va incrementando de forma poco más o menos constante durante la hinca de tubos y se determina sirviéndose de la medición de distancia electrónica dentro de la estación láser.

Por tanto, mediante el conocimiento de la posición absoluta de la estación láser se puede determinar también la posición absoluta y la dirección de la diana, y por consiguiente, se puede deducir la posición y la dirección de la máquina.

Esta información se combina con el curso del trazado indicado para el avance con hinca de tubos, para ofrecer así al conductor información clara e inequívoca acerca del lugar donde se encuentra la máquina en relación a la posición nominal. El sistema no sólo muestra la posición exacta de la máquina tuneladora en cualquier momento, sino que también prepara los resultados de forma clara y concluyente para el operador, de forma que éste pueda realizar las medidas de conducción correctivas que sean necesarias. Aparte de un enorme ahorro de tiempo, el sistema permite por lo tanto, también un avance con hinca de tubos uniforme y sin complicaciones próximo al eje nominal.

En avances con hinca de tubos es muy importante evitar cambios repentinos de dirección, y de esta forma limitar las cargas puntuales en los cantos exteriores del tubo, de ese modo se han de evitar daños a las juntas. Dichos daños no pueden tolerarse, ya que la penetración de agua subterránea en el tramo de tubería o la salida de aguas residuales al terreno circundante no debe producirse bajo ningún motivo.

Los cambios abruptos de dirección también traerían como consecuencia una gran resistencia de rozamiento contra el prensado previo del tramo completo de tubería, el cual solo podría superarse mediante cargas de presión especialmente elevadas.

Tan pronto como la máquina tuneladora y el tramo de tubería estén tan adelantados que el rayo láser ya no pueda alcanzar la diana desde el pozo de ataque, el láser deberá montarse a una consola, en una posición apropiada detrás de la diana, fijamente en el tramo de tubería.

2.1.2 Características del sistema de guiado de hinca de tubos

- Determinación de la posición de la máquina tuneladora con indicación en forma gráfica y numérica.
- Determinación e indicación de las tendencias de la máquina tuneladora.
- Reconocimiento inmediato de una desviación de la máquina.
- Cálculo de una curva de corrección, para reconducir la máquina tuneladora de forma tangencial al eje nominal.
- Información óptima de la posición de la máquina tuneladora al instante, para que el operario pueda conducirla lo más cerca posible del eje diseñado.
- Mayor contenido de información para un mando óptimo de la máquina tuneladora.
- Almacenamiento continúo de todos los datos de avance geométricos para la documentación.
- Indicación de las extensiones de los cilindros previamente calculadas para seguir una curva de corrección óptima.
- Control automático de dirección de la orientación actual del láser teodolito.
- Software de fácil manejo y hardware resistente y apto para túneles.
- Enorme ahorro de tiempo en comparación con los procedimientos convencionales.
- Incidir lo menos posible en la actividad de la obra para evitar pérdidas de tiempo.
- Control completo de todos los componentes del sistema desde un ordenador.
- Visualización de la posición de la máquina en un ordenador externo (opcional).
- Asistencia a escala mundial mediante enlace vía módem y telemantenimiento (opcional).

2.2 Análisis de las técnicas actuales y situación actual de la obra subterránea

Los conductores de las máquinas tuneladoras requieren recibir constantemente información referente a la situación actual del eje de la máquina, como desviaciones y tendencias, en relación al eje nominal. Con las velocidades de avance que actualmente se alcanzan de varios centímetros por minuto, el conductor de la máquina tuneladora precisa de una confirmación inmediata de los efectos de sus maniobras de conducción, de forma tal que pueda mantener la máquina tuneladora dentro de lo posible en el eje nominal.

El sistema de guiado con determinación automática de posición ofrece al conductor de la máquina tuneladora información constante sobre la posición y las tendencias de la máquina tuneladora. De esta forma se puede mantener la máquina tuneladora en el eje nominal sirviéndose de medidas de conducción equilibradas dentro de un radio de tolerancia relativamente reducido.

Para la ejecución correcta de un túnel desde el punto de vista topográfico es necesario que se cumplan los siguientes requisitos:

- Seguir la alineación especifica exigida por el cliente.
- Obtener una buena precisión en la realización del trabajo, lo cual implica una buena precisión en la salida de la máquina tuneladora, y una buena precisión en todo el trayecto de la misma.

Para ello es necesario disponer de un sistema de guiado que:

- Establezca unas referencias temporales para que el operador de la máquina tuneladora pueda seguir como referencia en la perforación del túnel.
- Presente la información de tal manera que el operario de la máquina tuneladora pueda conducirla lo más precisamente posible.

En la actualidad, debido a los grandes rendimientos que ofrecen las máquinas tuneladoras, alrededor de 30 metros de perforación al día, es necesario:

- Tener sistemas de guiado que ofrezcan la información de la posición de la máquina tuneladora al instante, para que el operario pueda conducirla lo más cerca del eje diseñado posible.
- Sistemas de guiado que incidan lo menos posible en la actividad de los trabajadores, para evitar pérdidas de tiempo en el trabajo de éstos.

Debido a todos los requisitos anteriormente mencionados, hace unos años que se descartó la topografía clásica para el replanteo de túneles realizados mediante máquinas tuneladoras, ya que los inconvenientes que se presentaban eran:

- Lentitud en la presentación de la información.

- Falta de visibilidad para un replanteo eficaz.

Por estas razones, se necesitaba de un sistema de guiado que incorporara a los conceptos clásicos de replanteo, los sistemas más modernos de cálculo por ordenadores, láseres, estaciones servomotorizadas, giróscopos, etc., consiguiendo sistemas de guiado que:

- Presenten información al instante de la posición de la máquina tuneladora con relación al eje diseñado.

- Ofrezcan una precisión excelente, normalmente por encima de las capacidades de conducción de las máquinas tuneladoras. Los túneles suelen realizarse con algunos centímetros de error en el trazado, pero son debidos a la imposibilidad de conducir la máquina tuneladora exactamente por el eje indicado por el sistema de guiado.

2.3 Componentes de un sistema de guiado

Los componentes del sistema de guiado que integran el hardware del sistema de guiado son los siguientes:

- Una diana sensible al láser montada en la máquina tuneladora como unidad de puntería activa, con un prisma para la medida electrónica de distancias montado en el mismo eje vertical del centro de la diana. Véase figura 2.

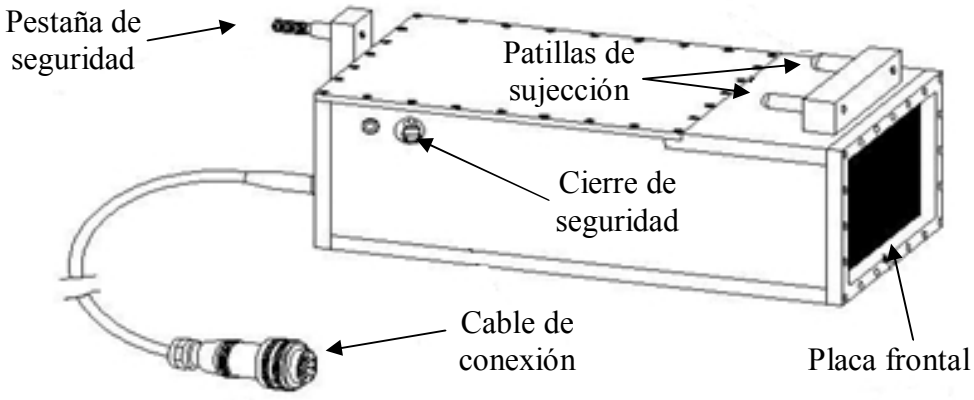


Figura 2. Componentes de la diana

- Una estación total servomotorizada con rayo láser integrada. En este caso se utiliza la estación total Leica TCA1200. Se estaciona en plataformas fijas situadas en el lateral del túnel, en lo que se denomina ventana del láser. En el diseño de las tuneladoras se tiene muy en cuenta la ventana del láser que es el espacio que hay que dejar libre de obstáculos para que el láser pueda incidir desde el teodolito hasta la diana. La estación total incluye un diodo láser integrado que está montado paralelamente al eje visual. El teodolito está equipado con un sistema sensor (ATR1) que permite la puntería automática de los prismas. Véase figura 3.

Figura 3. Vista de la estación Leica TCA 1200

- Un prisma que actúa como puntería anterior de referencia situada en la plataforma anterior en la que se situó la estación total láser. Sirve como visualización de la orientación de la estación total. La distancia entre las plataformas suele estar entre 30 y 100 metros que es la distancia mínima que se aconseja instalar la plataforma de la diana. Así mismo, la distancia máxima que se aconseja para que los errores de refracción del rayo láser no afecten en la precisión es de 100 metros. Véase figura 4.

Figura 4. Vista de la plataforma donde se ubica la puntería anterior de referencia

- Yellow Box. Sirve para el abastecimiento de corriente para el teodolito y el láser. Igualmente se organiza aquí la comunicación entre el ordenador y el teodolito.

- El equipo informático, fundamentalmente compuesto por un ordenador para la realización de los cálculos y para mostrar, a cada instante, la posición de la tuneladora en relación con el eje del túnel diseñado. Varias unidades de control electrónicas que permiten la transmisión de los datos entre el ordenador, la estación total y la diana, así como para poder obtener los datos de elongación de los cilindros de empuje del ordenador que controla todos los elementos de la tuneladora. Véase figura 5.

Figura 5. Vista de la cabina de control

- Caja central. La unidad de control establece la conexión a los sensores del sistema, y convierte su señal de salida a un formato apropiado para la alimentación en el ordenador de guiado. Igualmente en la unidad de control se convierten las señales de guiado del ordenador en sentido contrario y se transmiten a los sensores por separado.

- Tambor de cable. Con cada avance de la máquina tuneladora se prolonga la distancia entre el láser teodolito, el cual se encuentra fijo a la pared del túnel, y los restantes aparatos de medición, los cuales se encuentran en la máquina tuneladora misma o en los remolques. Ya que entre estas dos partes del sistema se halla un cable de datos, se debe contar con un dispositivo que permita las distancias variables sin que se ocasionen daños. La forma más confiable para la prolongación del cable es mediante un tambor de cable con contactos deslizantes.

- Módem. El módem se requiere para propósitos de telemantenimiento o asistencia. Se encuentra conectado a la red telefónica regular de la obra. De forma opcional también se puede indicar en cada caso, la posición actual de la máquina tuneladora en un ordenador localizado en la oficina de la obra.

- Otros aparatos. Por cada dos, un trípode sin plomada óptica, prisma estándar con soportes de prismas, base para los soportes de prismas y pernos adaptadores para los puntos de referencia del eje de la máquina.

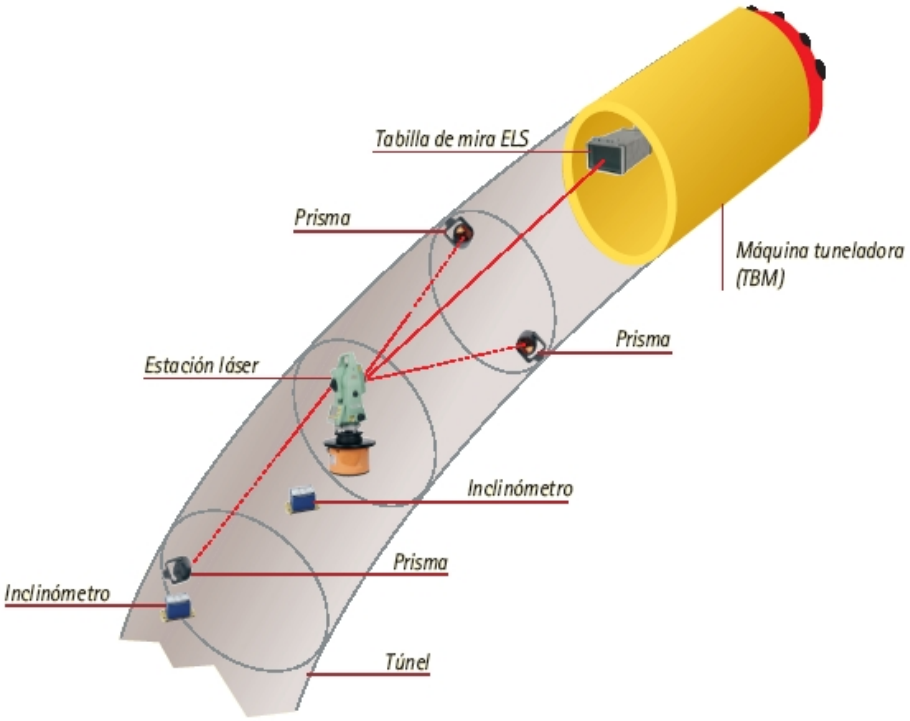


Figura 6. Componentes del sistema de guiado

2.4 Puesta en marcha y procedimiento básico del sistema de guiado

El procedimiento de utilización de este sistema de guiado es muy sencillo, y se podrían resumir en los siguientes pasos:

1. **Medición de los parámetros de la tuneladora**

Para poder guiar la tuneladora hay que determinar una serie de parámetros que definen la geometría de la tuneladora y de sus componentes, así como la geometría del trazado. Entre los parámetros que hay que determinar se encuentran:

- Diámetro de la máquina tuneladora de la tuneladora.
- Longitud de la máquina tuneladora.
- Definir la posición exacta de la diana es probablemente el parámetro más importante ya que un error provocaría desvíos en el guiado.
- Definir la posición y el número de gatos de empuje.
- Definir la posición y el número de los cilindros de articulación.
- Para poder obtener las lecturas de longitud de los gatos de empuje y de los cilindros de articulación hay que conseguir la comunicación entre el ordenador del sistema de guiado y el ordenador de la máquina tuneladora.
- En el caso de utilizar dovelas en la construcción del revestimiento del túnel, definir el número de dovelas y el tipo de dovelas que se van a emplear en la construcción de los anillos del túnel.
- Geometría del trazado del túnel.

2. Instalación del cableado y de los componentes del sistema de guiado

Una vez obtenidos todos los parámetros anteriores hay que introducirlos en el ordenador del sistema de guiado e instalar todos los componentes y el cableado que permite la comunicación entre todos ellos. En la figura 7 se pueden observar el cableado y los distintos componentes que intervienen en el guiado.

- Diana.
- Prisma.
- Unidad de control.
- Medidor de longitud de los cilindros de articulación.
- Medidor de longitud de los gatos de empuje.
- Estación total láser.
- Puntería anterior de referencia.

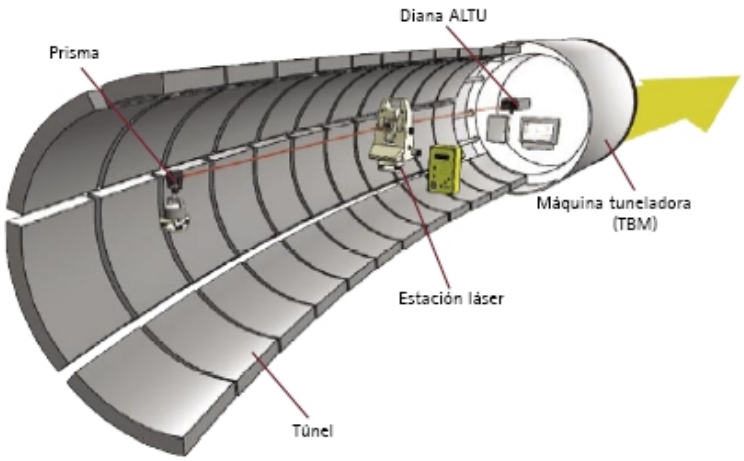


Figura 7. Vista de los componentes del sistema de guiado

3. Puesta en marcha del sistema de guiado

Una vez que se han conectado todos los componentes e introducido todos los parámetros necesarios para la configuración del sistema de guiado, se procede a la puesta en marcha.

El sistema de guiado, introduciendo solo las coordenadas de la estación total láser y del prisma anterior de referencia, el sistema de guiado será capaz de calcular la posición de la máquina tuneladora y en consecuencia las desviaciones verticales, horizontales y desviaciones angulares respecto con el eje del trazado.

4. Cambios de consola

Con el avance de la máquina tuneladora llegará un momento que la diana estará demasiado lejos para una correcta observación de la misma desde la plataforma de la estación total láser. Con lo que habrá que adelantar la posición de la estación sobre una nueva plataforma e introducir las nuevas coordenadas de la nueva posición del láser. Esta operación se hará sucesivamente hasta la finalización del túnel.

Figura 8. Vista de la plataforma donde se ubica la puntería anterior de referencia 2

En la figura 8, se muestra la imagen de la plataforma donde se ha ubicado la puntería de referencia anterior de la estación láser. Ésta ha concluido su función ya que el láser está demasiado alejado de la diana sensible montada en la máquina tuneladora.

2.5 Parámetros a mostrar en un sistema de guiado

En la actualidad, cada empresa dedicada a la realización de túneles a través de máquina tuneladora dispone de su propio software de guiado. Es decir, una interfaz gráfica externa que permite monitorizar los parámetros más significativos para el correcto guiado y la representación de la trayectoria de su posición espacial.

A continuación, se muestran algunas de las interfaces gráficas que han utilizado diferentes empresas de construcción, para la monitorización de la posición de la máquina tuneladora.

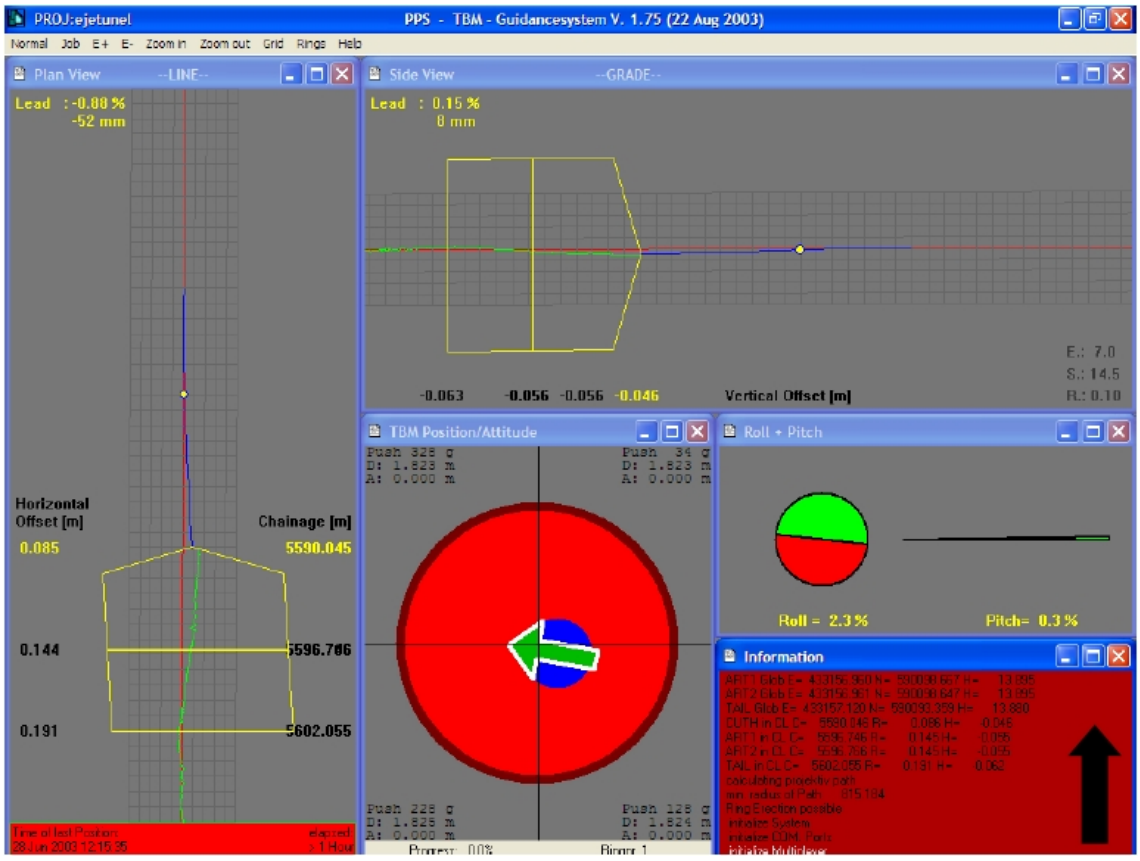


Figura 9. Vista de la pantalla principal del sistema de guiado PPS - TBM

Como se aprecia en la figura 9, esta interfaz gráfica permite una visualización de los tres puntos de vista planta, alzado y perfil de la máquina tuneladora. De esta manera, se pueden apreciar las desviaciones verticales, horizontales de su trayectoria ideal.

Otra ventana, permite apreciar los giros que se producen sobre el eje longitudinal "Roll" o alabeo y los giros que se producen sobre el eje transversal "Pitch" o inclinación.

Por último, se muestra un panel informativo donde se monitorizan los parámetros necesarios para poder guiar la máquina tuneladora.

En la siguiente imagen se muestra otro prototipo de interfaz gráfica. Véase figura 10.

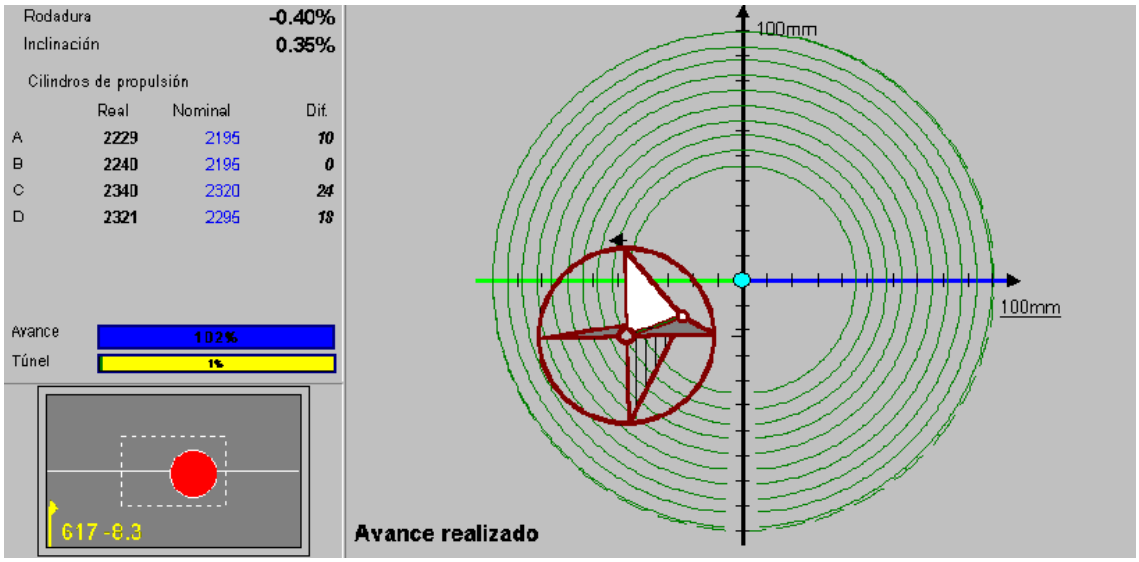


Figura 10. Vista de la pantalla principal del sistema de guiado TBM

Esta interfaz gráfica monitoriza la posición de la máquina tuneladora mediante una flecha simbólica y representa unos círculos concéntricos que indican la dirección hacia donde se dirige la trayectoria ideal del túnel.

En la parte izquierda del panel de control se muestran parámetros técnicos que indican las desviaciones angulares que siguen la trayectoria de la máquina y la presión que están ejerciendo los cilindros hidráulicos sobre el cabezal de corte de la máquina.

También representa una barra de avance donde indica el porcentaje total para la finalización del túnel y otra barra que indica el porcentaje actual en que se encuentra el proceso de tunelación.

El software más moderno que actualmente se está empleando para la representación gráfica de la posición y trayectoria que recorre una máquina tuneladora ha sido diseñado por la empresa alemana VMT. Este programa llamado SLS-Microtunnelling LT se divide en varias versiones diferentes que han sido desarrolladas específicamente para la construcción de cada tipo de túnel.

Para más información acerca de la empresa alemana VMT y el sistema de guiado SLS-Microtunnelling LT, véase referencia a su página web en bibliografía [8].

Otra versión utilizada para la monitorización de la construcción de túneles por el método de hinca de tubos se observa en la figura 11.

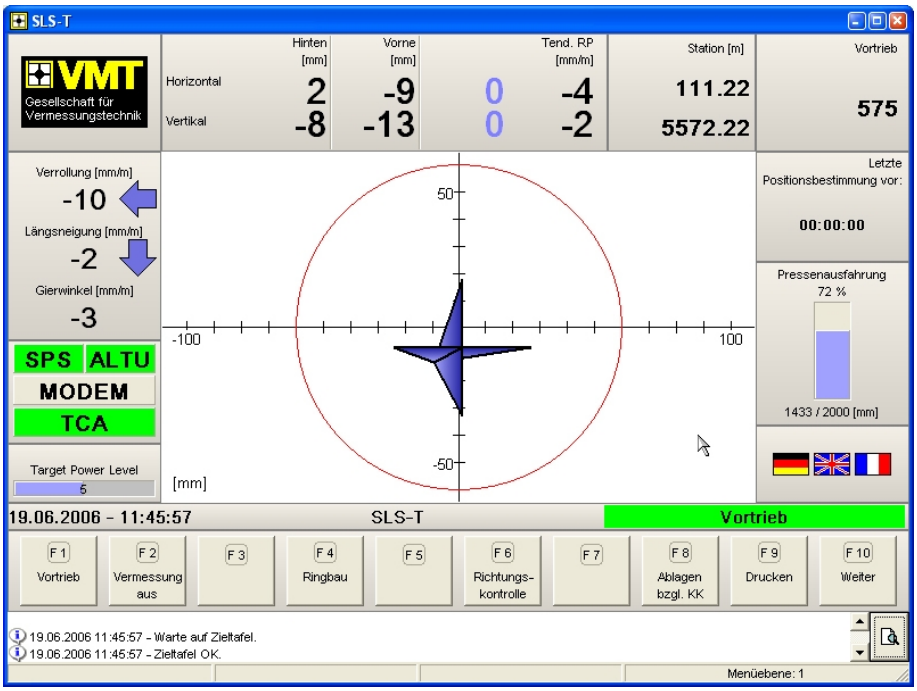


Figura 11. Vista de la pantalla principal del sistema de guiado SLS-T

La versión actualmente utilizada muestra algunos parámetros técnicos que la anterior versión no cuenta pero es menos intuitiva para el correcto guiado de la máquina tuneladora. Véase figura 12.

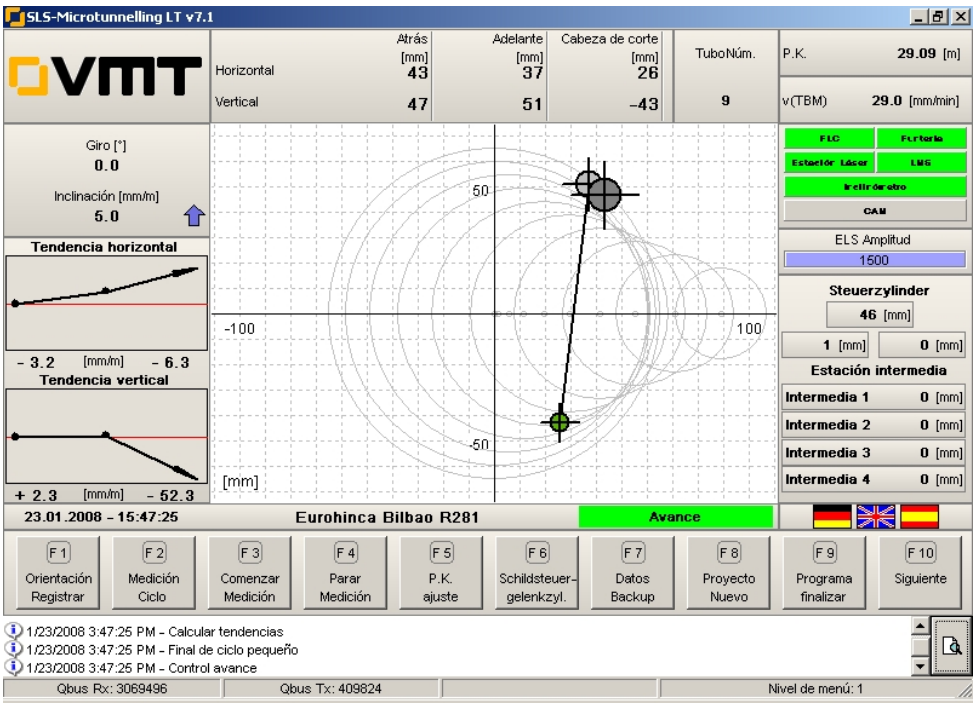


Figura 12. Vista de la pantalla del sistema de guiado actual SLS-Microtunnelling

A continuación, van a describirse detalladamente los parámetros de guiado que son comunes de ambas versiones.

- *Coordenadas horizontales y verticales*

  Son las coordenadas de posición de la parte delantera y trasera de la máquina tuneladora. Indican los desplazamientos verticales y horizontales de ambas posiciones entre el eje de proyecto y el eje de la máquina tuneladora. Estas desviaciones se dan cada vez que se toma una referencia de posición, puesto que al dar coordenadas de dichos puntos, se comparan con el trazado que está almacenado en memoria. Los valores pueden ser positivos y negativos, y su significado en cada caso es:

  – *Desviación horizontal*
  Componente horizontal de la distancia entre la máquina tuneladora y el eje de proyecto. Si su valor es positivo, la máquina está a la derecha del eje. Si es negativo, está a la izquierda.

  – *Desviación vertical*
  Componente vertical de la distancia entre la máquina tuneladora y el eje de proyecto. Si su valor es positivo, la máquina está por encima del eje. Si es negativo, está por debajo. Véase figura 12.

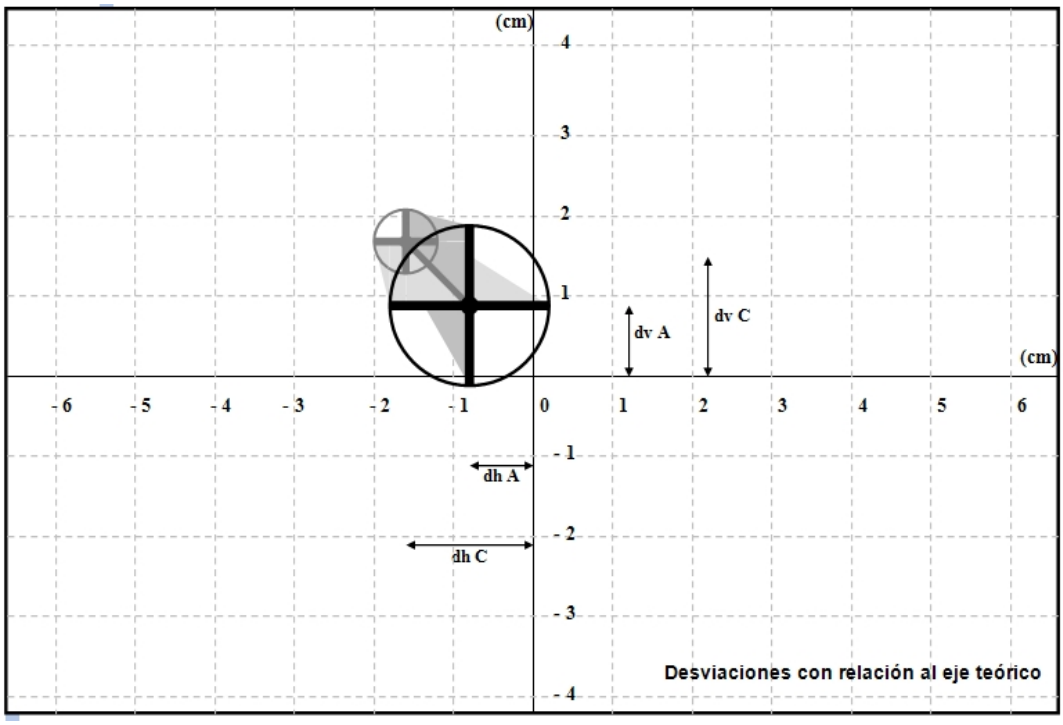


Figura 13. Desviaciones horizontales y verticales de la tuneladora

- ***Desviaciones angulares***

Son valores que indican los tres giros de la máquina tuneladora respecto del eje teórico del trazado. Si se produce un giro en el eje vertical respecto del sistema de referencia de la misma, se le denomina guiñada (yaw). Si se produce un giro en el eje horizontal se denomina cabeceo (pitch). Por último, si se produce un giro en el eje situado de la parte delantera a la parte trasera de la máquina tuneladora se denomina alabeo (roll). Véase figura 14.

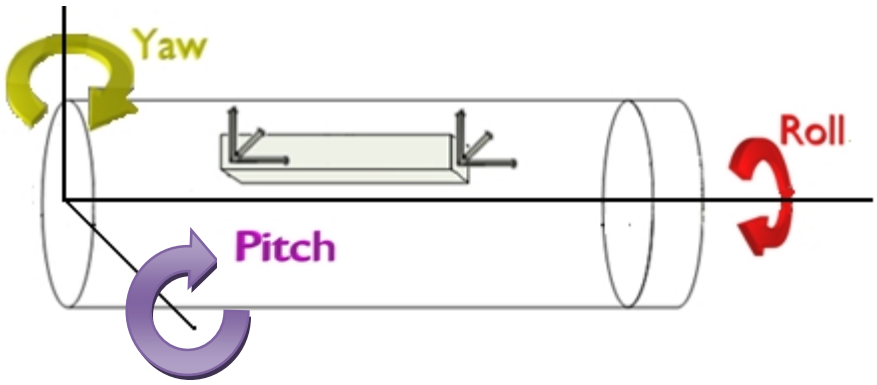


Figura 14. Rotaciones angulares de la tuneladora

En la interfaz gráfica, estos giros de la máquina tuneladora sobre su sistema de referencia se denominan inclinación (cabeceo), incidencia (guiñada) y giro (alabeo).

– *Incidencia*

Es el ángulo que rota la máquina tuneladora con respecto al eje vertical. La incidencia es positiva cuando la máquina tuneladora queda a la derecha del eje longitudinal y negativa en caso contrario. Véase figura 15.

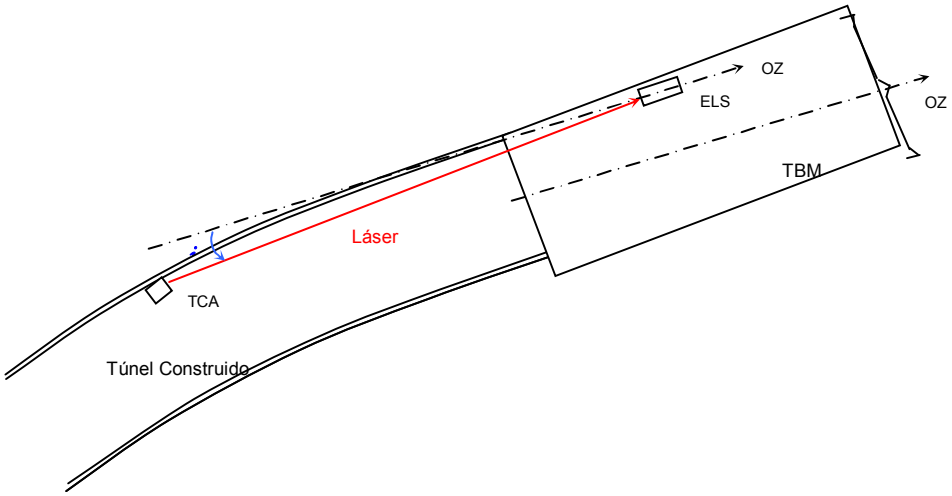


Figura 15. Ángulo de incidencia. Vista en planta

– *Inclinación*

Es el ángulo que rota la máquina tuneladora con respecto al eje transversal. La inclinación es positiva cuando la máquina tuneladora está por encima del horizonte y negativo en caso contrario. Véase figura 16.

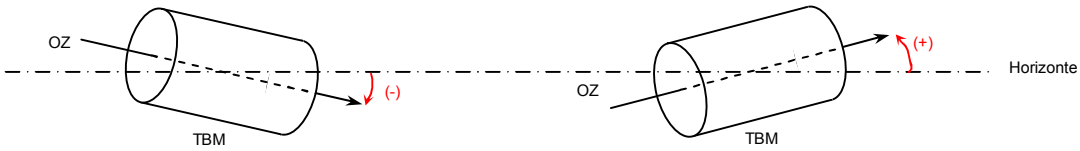


Figura 16. Ángulo de inclinación. Secciones longitudinales

– *Giro*

Es el ángulo que forma el eje OX de la máquina tuneladora con un plano horizontal, medido en el plano vertical que contiene a dicho eje. Será positivo cuando, en el sentido del avance, ésta quede por encima del horizonte, y negativo cuando quede por debajo. Véase figura 17.

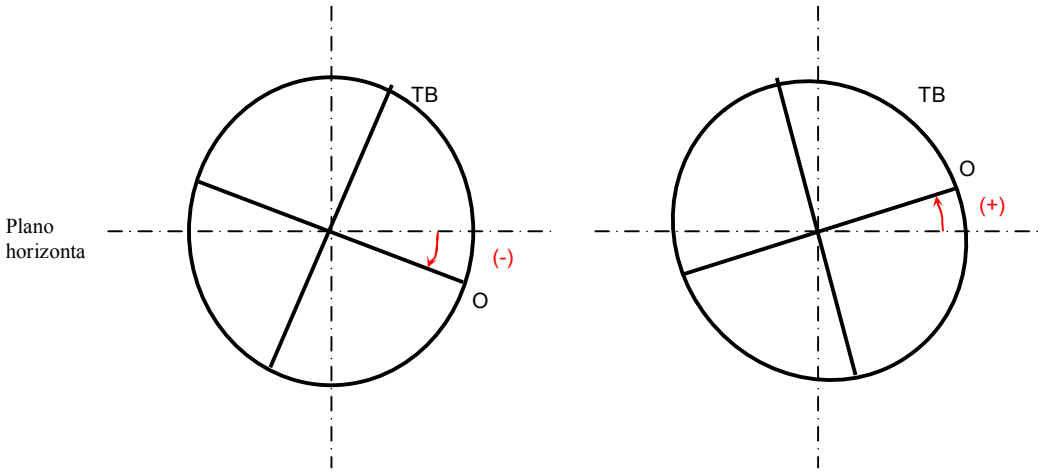


Figura 17. Ángulo de giro. Sección transversal

- ***Tendencias***

Las tendencias indican el movimiento que tendrá la máquina tras un avance, dando una idea del alcance de la corrección que se le está aplicando. El piloto va guiando la máquina ejerciendo ciertas presiones en los cilindros de guiado y en los de empuje y esos valores de presión los toma el programa para calcular las tendencias. Gracias a este cálculo de la trayectoria futura que está recorrerá la máquina, el operario puede anticiparse y realizar las pertinentes modificaciones para aproximar la trayectoria real de una manera más precisa a la trayectoria ideal.

– *Tendencia horizontal*
  Coordenada horizontal donde se situará la máquina tuneladora en el siguiente avance. Si su valor es positivo, la máquina estará a la derecha del eje. Si es negativo, estará a la izquierda.

– *Tendencia vertical*
  Coordenada vertical donde se situará la máquina tuneladora en el siguiente avance. Si su valor es positivo, la máquina estará por encima del eje. Si es negativo, estará por debajo.

- ***Punto kilométrico***

Valor que indica la distancia recorrida desde el comienzo de la tunelación hasta la parte delantera de la posición actual donde se encuentra la máquina tuneladora. La distancia recorrida se expresa en metros.

- ***Avances***

Valor que indica el número de avances que se han producido durante todo el proceso de la tunelación, es decir, la cantidad de veces que el programa ha refrescado los parámetros a mostrar.

- ***Número de tubos***

Valor que indica la cantidad de tubos que se han hincado en el túnel durante el proceso.

- ***Velocidad de avance***

Valor que indica la velocidad que lleva la máquina tuneladora a lo largo de toda la excavación. La velocidad se expresa en mm/min.

- ***Amplitud del láser***

Indica en que porcentaje ha variado la intensidad de la señal del haz inicial respecto al recibido en la tarjeta.

- ***Intensidad del láser***

Indica el diámetro del haz láser recibido en la tarjeta a bordo de la máquina. El diámetro que muestra la intensidad del láser se expresa en mm.

- ***Hora y fecha***

Indica la fecha y hora actualizada de cada avance y representación gráfica que se produce en la máquina tuneladora.

2.6 Posibles mejoras de los sistemas de guiado

2.6.1 Nuevos láser para la hinca de tubería con máquina tuneladora

A día de hoy, los rayos láser se están utilizando como referencia para el sistema de guiado de hinca de tubería. Este rayo láser se alinea con la alineación del trazado tanto horizontal como verticalmente. Se alinea horizontalmente mediante marcas realizadas por topografía de precisión realizadas en las paredes del pozo de ataque y que marcan la alineación horizontal a seguir y verticalmente introduciendo la pendiente en el instrumento láser.

Sin embargo, en hincas de tubería de gran longitud, mayores de 350 metros, las diferencias de temperatura a lo largo del túnel y la cantidad de partículas del aire hacen que la refracción del rayo no siga una alineación totalmente recta, además de producirse una amplificación y desconcentración del rayo. Por tanto, se pierde precisión, lo que nos impide utilizar este rayo como referencia del sistema de guiado.

Se ha estudiado la utilización de rayos láser de otra longitud de onda, para los cuales, las condiciones extremas de una hinca de tubos con rozadora, diferencias de temperatura y visibilidad por el polvo resultante de la perforación, no afecten a la calidad y precisión del rayo láser.

Los equipos estándar de hinca utilizan rayos láser rojos con una longitud de onda de 633 nm y los estudiados actualmente están utilizando rayos láser verdes con una longitud de onda de 532 nm y se están observando mejorías en cuanto a la calidad y precisión del rayo. Esta innovación en rayo láser, se está utilizando en los equipos de escudo abierto.

En las máquinas tuneladoras de escudo cerrado los sistemas de guiado utilizan unas dianas sensibles a la intensidad del rayo láser captando su punto de mayor incidencia detectando su posición tanto en vertical como en horizontal. Con esta posición, como con su ángulo de inclinación e incidencia del rayo, el sistema de guiado es capaz de replantear la máquina en el espacio y se puede mostrar en el monitor del sistema de guiado sus desviaciones verticales horizontales y angulares. Véase figuras 18 y 19.

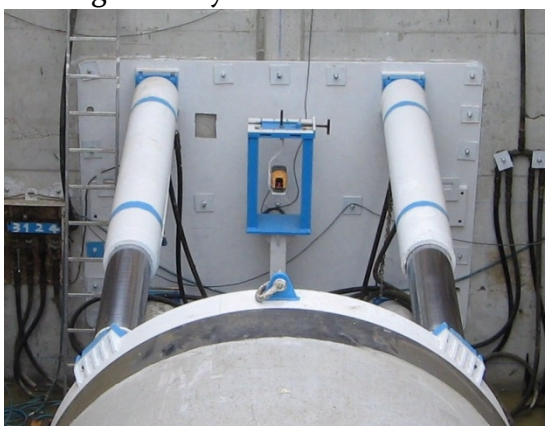


Figura 18. Vista del láser colocado en el pozo de ataque en una hinca de tubería

Figura 19. Vista del prototipo de láser

2.6.2 Diseño de un prototipo para una nueva diana láser

Para ello, habría que utilizar dos cámaras ya que para la obtención de los valores angulares hay que medir en dos planos distintos. El primer plano estaría materializado por un plano transparente pero en el que se marcaría el láser y el segundo plano sería opaco en el que también se marcaría el rayo láser. Ambos planos serían el plano frontal y el plano posterior. Estos planos deben estar marcados con una cuadricula milimétrica con el fin de obtener una imagen del plano y del la marca del rayo láser y poder definir el centro del rayo láser con precisión milimétrica. Estás dianas tendrían dos ventajas respecto a las dianas actuales.

En primer lugar podrían utilizar un rayo láser más potente, el rayo láser verde, con lo que se podría hacer hincas de más distancia con la misma precisión a la actual.

En segundo lugar, las dianas actuales son de reducidas dimensiones ya que el coste de las células fotosensibles es muy importante, por lo que se tiene una ventana del láser muy pequeña. Así que, en el caso de desviaciones de la máquina donde se pueda producir la pérdida de la ventana del láser, hay que cambiar la posición de éste para colocarlo de nuevo centrado en la ventana del láser. Por el mismo coste se podría construir una diana con cámaras ópticas del doble de dimensiones.

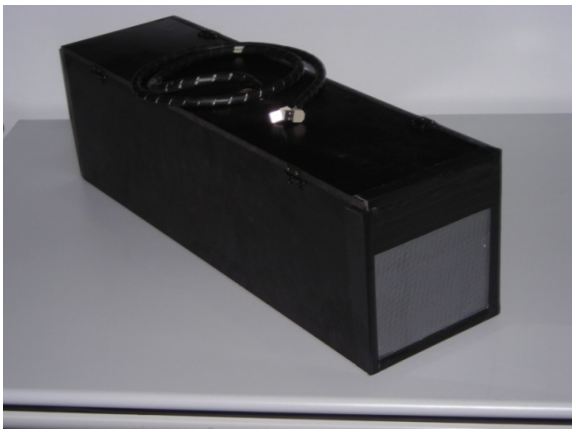


Figura 20. Vista del exterior del prototipo de diana

Figura 21. Vista del interior del prototipo de diana

En las figuras 20 y 21, se muestra el prototipo de diana sensible que se ha utilizado para desarrollar el sistema de guiado de la máquina tuneladora. Además, se ha programado un software para la estimación de las coordenadas de posición de la máquina tuneladora, gracias a la ayuda de inclinómetros y cámaras de vídeo para el cálculo paramétrico.

A continuación, se muestra un esquema gráfico del interior de la diana sensible donde se pueden apreciar la colocación de todos los componentes citados anteriormente. Véase figura 22.

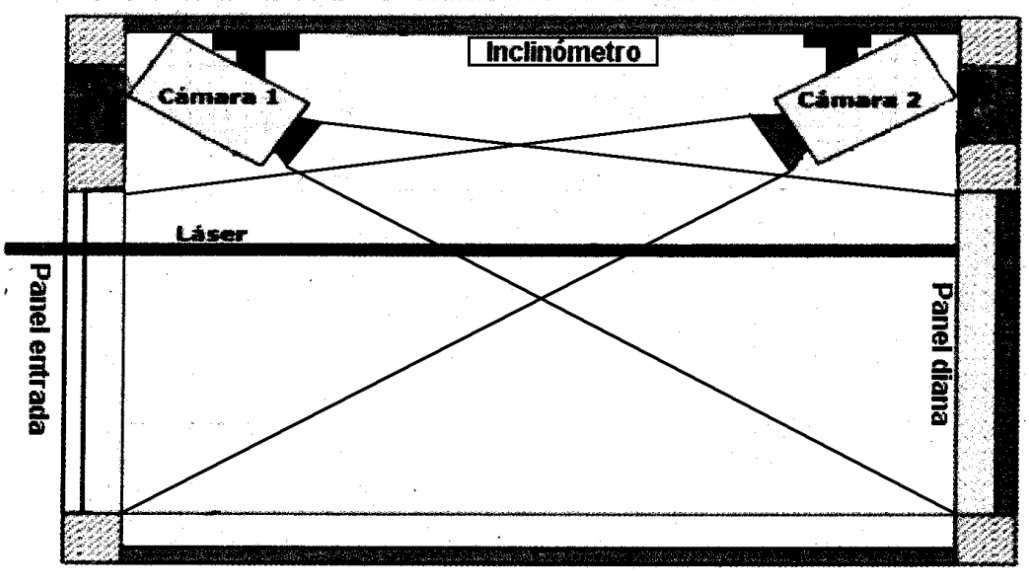


Figura 22. Vista en perpendicular de la disposición de los componentes del prototipo de una diana con cámaras

En primer lugar, para realizar la medición del ángulo de rodadura, se ha colocado en el interior de la diana un inclinómetro que mida el valor del ángulo del giro sobre el eje longitudinal de la máquina tuneladora.

En segundo lugar, con los puntos donde incide el láser de los paneles delantero y trasero de la diana, se evalúa tanto la incidencia como el ángulo de inclinación. Además, se ha insertado otro inclinómetro para medir el valor de inclinación y contrastarlo con el valor calculado por la posición del láser en los planos delantero y trasero en la diana.

La implementación del sistema de la diana se ha desarrollado en El Proyecto de Fin de Carrera "Desarrollo de prototipo de diana para el guiado de una tuneladora" realizado por Rafael Portero Hernández.

2.7 Interfase Hombre-Máquina

La sigla HMI es la abreviación en inglés de Interfaz Hombre Máquina. Los sistemas HMI podemos pensarlos como una ventana de un proceso. Esta ventana puede estar en dispositivos especiales como paneles de operador o en una computadora. Los sistemas HMI en computadoras se los conoce también como software HMI o de monitoreo y control de supervisión. Las señales del proceso son conducidas al HMI por medio de dispositivos como tarjetas de entrada/salida en la computadora, PLC's (Controladores lógicos programables), RTU (Unidades remotas de I/O) o DRIVE's (Variadores de velocidad de motores). Todos estos dispositivos deben tener una comunicación que entienda el HMI. Véase figura 23.

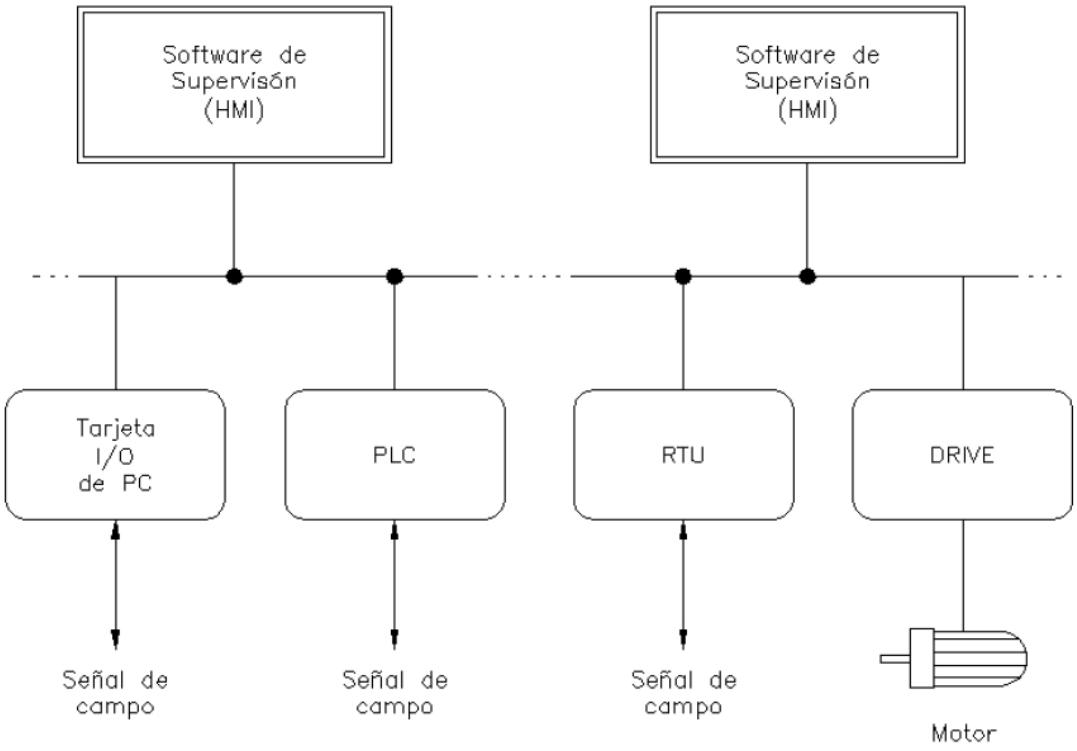


Figura 23. Sistema HMI

La estructura de los software's HMI está compuesta por un conjunto de programas y archivos. Hay programas para diseño y configuración del sistema y otros que son el motor mismo del sistema. En la figura 24 se muestra cómo funcionan algunos de los programas y archivos más importantes. Los rectángulos de la figura representan programas y las elipses representan archivos. Los programas que están con recuadro simple representan programas de diseño o configuración del sistema. Los que tienen doble recuadro representan programas que son el motor del HMI.

Con los programas de diseño, como el "Editor de pantallas" se crea moldes de pantallas para visualización de datos del proceso. Estos moldes son guardados en archivos "Archivo de pantalla" y almacenan la forma como serán visualizados los datos en las pantallas.

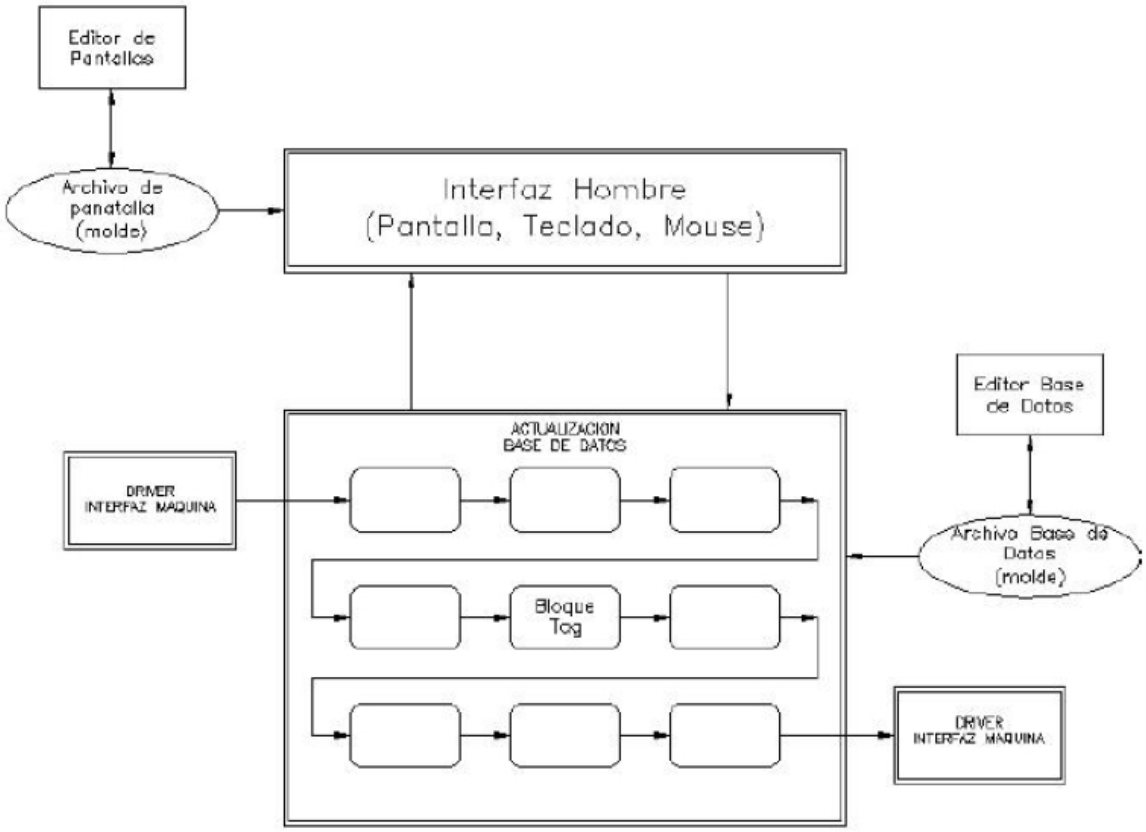


Figura 24. Estructura general del software HMI

Algunas de las funciones de un software HMI son:

- Monitoreo. Es la habilidad de obtener y mostrar datos de la planta en tiempo real. Estos datos se pueden mostrar como números, texto o gráficos que permitan una lectura más fácil de interpretar.

- Supervisión. Esta función permite junto con el monitoreo la posibilidad de ajustar las condiciones de trabajo del proceso directamente desde la computadora.

- Alarmas. Es la capacidad de reconocer eventos excepcionales dentro del proceso y reportarlo estos eventos. Las alarmas son reportadas basadas en límites de control preestablecidos.

- Control. Es la capacidad de aplicar algoritmos que ajustan los valores del proceso y así mantener estos valores dentro de ciertos límites. Control va más allá del control de supervisión removiendo la necesidad de la interacción humana. Sin embargo la aplicación de esta función desde un software corriendo en una PC puede quedar limitada por la confiabilidad que quiera obtenerse del sistema.

- Históricos. Es la capacidad de mostrar y almacenar en archivos, datos del proceso a una determinada frecuencia. Este almacenamiento de datos es una poderosa herramienta para la optimización y corrección de procesos.

Algunas de las tareas de un software HMI de supervisión y control son:

- Permitir una comunicación con dispositivos de campo.

- Actualizar una base de datos dinámica con las variables del proceso.

- Visualizar las variables mediante pantallas con objetos animados (mímicos).

- Permitir que el operador pueda enviar señales al proceso, mediante botones, controles ON/OFF, ajustes continuos con el mouse o teclado.

- Supervisar niveles de alarma y alertar/actuar en caso de que las variables excedan los límites normales.

- Almacenar los valores de las variables para análisis estadístico y/o control.

- Controlar en forma limitada ciertas variables de proceso.

2.8 Interfaces gráficas y entornos de programación gráfica

Con la idea de simplificar el uso de los ordenadores para usuarios de todo tipo y no sólo para los expertos, se ha convertido en una práctica habitual utilizar metáforas visuales por medio de la llamada interfaz gráfica de usuario (IGU ó GUI en inglés) para que el usuario interactúe y establezca un contacto más fácil e intuitivo con el ordenador. En estos casos, un simple clic de ratón sobre algún gráfico que aparece en la pantalla, sustituye a la tediosa tarea de escribir código fuente para que el ordenador interprete que debe realizar alguna acción. En 1981 aparecieron los primeros ordenadores personales, los llamados Pc's, pero hasta 1993 no se generalizaron las interfaces gráficas de usuario. El escritorio del sistema operativo Windows de Microsoft y su sistema de ventanas sobre la pantalla se ha estandarizado y universalizado, pero fueron los ordenadores Macintosh de la compañía Apple los primeros que introdujeron las interfaces gráficas de usuario.

Una interfaz es un dispositivo que permite comunicar dos sistemas que no hablan el mismo lenguaje. Restringido a aspectos técnicos, se emplea el término interfaz para definir el juego de conexiones y dispositivos que hacen posible la comunicación entre dos sistemas. Sin embargo, cuando aquí se habla de interfaz se referire a la cara visible de los programas tal y como se presenta a los usuarios para que interactúen con la máquina. La interfaz gráfica implica la presencia de un monitor de ordenador o pantalla constituida por una serie de menús e iconos que representan las opciones que el usuario puede tomar dentro del sistema.

La interfaz proporcionará al usuario el conjunto de posibilidades que podrá seguir durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que puede realizar, así como las respuestas que puede ofrecer el sistema. El usuario, además de entender el mensaje, ha de comprender la mecánica operativa que se le ofrece (sintaxis, órdenes, códigos, abreviaturas, iconos, etc.). Una buena interfaz requiere poco esfuerzo por parte del usuario, simplicidad y funcionalidad.

Las características básicas de una buena interfaz podrían sintetizarse en:

- Facilidad de comprensión, aprendizaje y uso.
- Representación fija y permanente de un determinado contexto de acción (fondo).
- El objeto de interés ha de ser de fácil identificación.
- Diseño ergonómico mediante el establecimiento de menús, barras de acciones e iconos de fácil acceso.
- Las interacciones se basarán en acciones físicas sobre elementos de código visual o auditivo (iconos, botones, imágenes, mensajes de texto o sonoros, barras de desplazamiento y navegación...) y en selecciones de tipo menú con sintaxis y órdenes.

- Las operaciones serán rápidas, incrementales y reversibles, con efectos inmediatos.
- Existencia de herramientas de Ayuda y Consulta.
- Tratamiento del error bien cuidado y adecuado al nivel de usuario.

La tipografía y el tratamiento del color son dos elementos a los que hay que prestar especial importancia a la hora de establecer una buena interfaz, poniendo especial cuidado en el diseño de las formas y la coherencia interna entre ellas.

Para diseñar una buena interfaz enfocada hacia el usuario es necesario tener claros los objetivos del hipertexto, teniendo en cuenta no sólo lo que se persigue ofreciendo información, sino las necesidades que van a tener los usuarios a la hora de consultarlo. También es clave determinar el contenido y la funcionalidad, especificar la estructura organizativa, la navegación, las secciones y los sistemas de búsqueda. Hay que tener en cuenta que cada usuario puede tener diferentes necesidades y un buen sistema de navegación debe contar con las herramientas adecuadas para diferentes funciones. Como cada usuario pude tener diferentes necesidades, es importante ofrecer diferentes formas de acceso y búsqueda, desde búsquedas precisas, hasta exploraciones guiadas o a elección del lector.

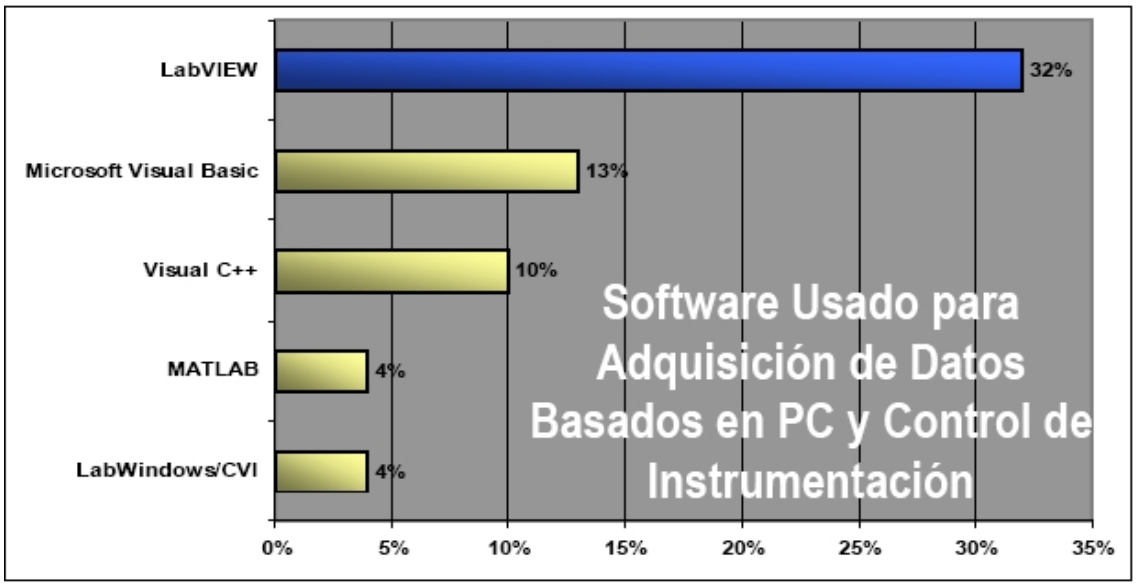


Figura 25. Comparación de Software para el desarrollo de HMI's

Los lenguajes de programación visual como Visual C++, Visual Basic, LabVIEW, MATLAB, etc., son entornos de programación gráfica y se utilizan para desarrollar software HMI a medida del usuario, es decir, interfaces gráficas de usuario. Véase figura 25.

A continuación, se exponen algunos de los más importantes entornos de programación gráfica utilizados para la creación de GUI's.

2.8.1 MATLAB (GUIDE)

Para el desarrollo de la aplicación en tiempo real, se trabaja con una herramienta de MATLAB llamada GUIDE (Graphical Use Interface Development Environment). Esta herramienta esta pensada para desarrollar GUI's (Graphical User Interfaces) fácil y rápidamente haciendo sencillo el diseño y presentación de los controles de la interfaz, reduciendo la labor en el momento de seleccionar, deshacer, arrastrar y centrar controles, así como la personalización de las propiedades de estos.

El proceso a seguir para el desarrollo de un programa mediante GUIDE es que una vez se tienen todos los controles en posición, se editan las funciones de llamada (Callback) de cada uno de ellos, escribiendo el código de MATLAB que se ejecutará cuando el control sea utilizado. GUIDE está diseñado para hacer menos tedioso el proceso de desarrollo de la interfaz gráfica, para ello cuenta con un editor de propiedades (property editor) con el que se podrá modificar en cualquier momento los nombres, valores por defecto y las propiedades de los elementos.

Encontraremos doce herramientas con las que podemos crear nuestra interfaz: Push Button, Toggle Button, Radio Button, CheckBox, Edit Text, Static Text, Slider, Table, Panel, ListBox, Popup Menu, Axes. Así mismo, encontraremos un formulario en blanco sobre el cual podremos empezar a construir el nuestro proyecto.

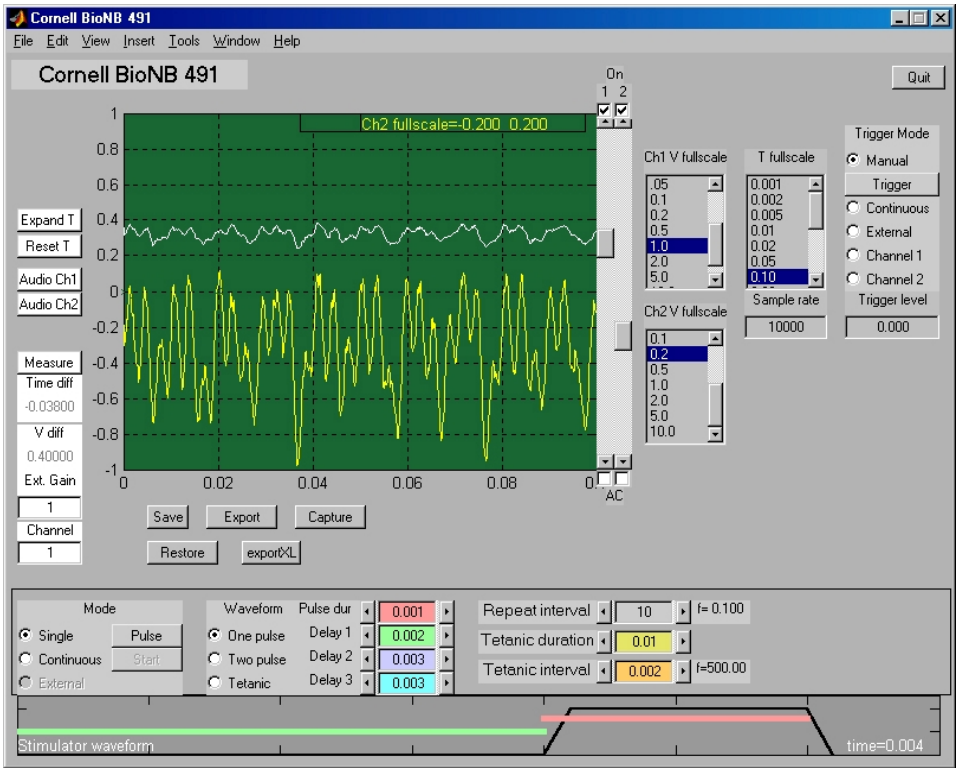


Figura 26. Interfaz gráfica en MATLAB

Gracias a toda esta cantidad de múltiples propiedades gráficas, el usuario podrá realizar un diseño de GUI adaptado a sus necesidades y preocuparse sólo por la programación del software. Véase figura 26.

2.8.2 LabVIEW

Es una herramienta gráfica de programación, esto significa que los programas no se escriben, sino que se dibujan, facilitando su comprensión. Al tener ya pre-diseñados una gran cantidad de bloques, se le facilita al usuario la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interactuación con el usuario final. Cada hoja de trabajo consta de dos partes diferenciadas:

- **Panel Frontal**. Es la interfaz con el usuario, se utiliza para interactuar con el usuario cuando el programa se esta ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real y como van fluyendo los datos de entradas y salidas. En esta interfaz se definen los controles que se usan como entradas, pueden ser botones, marcadores, etc. También se definen los indicadores que se usan como salidas, pueden ser gráficas, matrices, etc.

- **Diagrama de Bloques**. Es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan iconos que realizan una determinada función y se interconectan con el código que controla el programa. Suele haber una tercera parte icono/conector que son los medios utilizados para conectar una hoja de trabajo con otras hojas de trabajo. Véase figura 27.

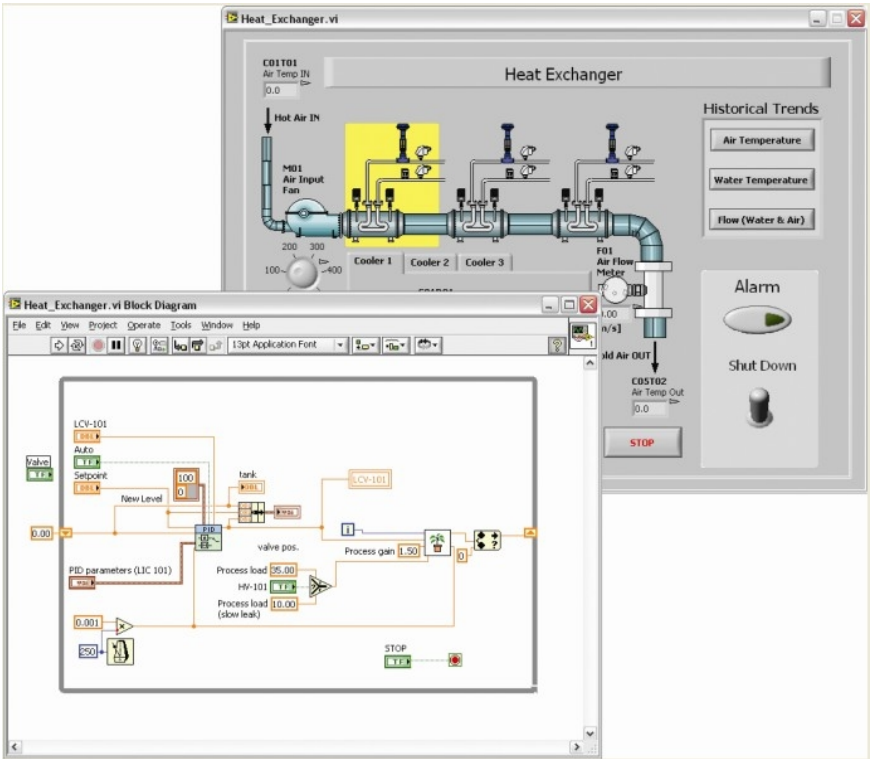


Figura 27. Interfaz gráfica en LabVIEW

2.8.3 Delphi

Delphi es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual.
Un uso habitual de Delphi, aunque no el único, es el desarrollo de aplicaciones visuales y de bases de datos cliente-servidor y multicapas. Debido a que es una herramienta de propósito múltiple, se usa también para proyectos de casi cualquier tipo, incluyendo aplicaciones de consola, aplicaciones de web, servicios COM y DCOM y servicios del sistema operativo.

Como entorno visual, la programación en Delphi consiste en diseñar los formularios que componen al programa colocando todos sus controles (botones, etiquetas, campos de texto, etc.) en las posiciones deseadas, normalmente usando un ratón. Luego se asocia código a los eventos de dichos controles y también se pueden crear módulos de datos, que regularmente contienen los componentes de acceso a datos y las reglas de negocio de una aplicación.

Una de las principales características y ventajas de Delphi es su capacidad para desarrollar aplicaciones con conectividad a bases de datos de diferentes fabricantes. El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de código abierto o gratuito.

Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo.

Una gran parte de los componentes disponibles para Delphi son controles que encapsulan los elementos de interacción con el usuario como botones, menús, barras de desplazamiento, etc.

Además de poder utilizar en un programa estos componentes estándar (botones, grillas, conjuntos de datos, etc.), es posible crear nuevos componentes o mejorar los ya existentes, extendiendo la funcionalidad de la herramienta. En internet existe un gran número de componentes, tanto gratuitos como comerciales, disponibles para los proyectos a los que no les basten los que vienen ya con la herramienta.

2.8.4 Visual Basic

Visual Basic es un lenguaje de programación que fue desarrollado con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y en cierta medida también la programación misma.

Visual Basic constituye un IDE (entorno de desarrollo integrado o en inglés Integrated Development Enviroment) que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código (programa donde se escribe el código fuente), un depurador (programa que corrige errores en el código fuente para que pueda ser bien compilado), un compilador (programa que traduce el código fuente a lenguaje de máquina) y un constructor de interfaz gráfica o GUI (es una forma de programar en la que no es necesario escribir el código para la parte gráfica del programa, sino que se puede hacer de forma visual).

La ventana de propiedades contiene diferentes formas para utilizar el programa, las cuales son: (Pointer) Apuntador o puntero, (Label) Etiqueta, (Frame) Marco, (CheckBox) Casilla de verificación, (ComboBox) Lista desplegable, (HScrollBar) Barra de desplazamiento horizontal, (Timer) Temporizador, (DirListBox) Lista de directorios, (Shape) Figura, (Image) Imagen, (PictureBox) Caja de Imagen, (TextBox) Caja de texto, (CommandButton) Botón de pulsación, (OptionButton) Botón de opción, (ListBox) Lista, (VScrollBar) Barra de desplazamiento vertical, (DriveListBox) Lista de unidades de disco, (FileListBox) Lista de archivos, (Line) Línea y por último (Data) Datos.

Gracias a toda esta cantidad de múltiples propiedades gráficas, el usuario podrá realizar un diseño de GUI adaptado a sus necesidades y preocuparse sólo por la programación del software.

3. Desarrollo de la interfaz gráfica para el guiado

3.1 Especificaciones de diseño

Las actuales tuneladoras de excavación de túneles, son máquinas muy costosas y complejas que cuentan con un alto grado de sofisticación tecnológica. Sin embargo, es un hecho comprobado que su guiado no está optimizado.

En vista de las necesidades ocasionadas, se requiere realizar un sistema de guiado que sea completamente funcional y posea todas las características necesarias para llevar a cabo un preciso guiado de la trayectoria de una máquina tuneladora.

Una de las características fundamentales, es que el sistema de guiado tiene que tener un índice de repetitividad muy alto, para poder ofrecer al operario que monitoriza la interfaz gráfica, la posición espacial de ésta en tiempo real.

Por tanto, se ha diseñado un sistema de guiado basado en tres módulos funcionales, que deben estar correctamente implementados y comunicados para poder llevar a cabo su cometido. Este sistema ha utilizado nuevos componentes más sofisticados gracias a las labores de investigación y desarrollo realizadas previamente.

Los tres módulos que forman el sistema de guiado son:

- Diana de puntería láser.
- Interfaz gráfica de usuario (GUI).
- Simulador de hinca de tubos.

La diana de puntería láser será la encargada de calcular y ofrecer los valores reales de las coordenadas de posición de la máquina tuneladora durante el proceso de tunelación.

El simulador hinca de tubos realiza las simulaciones de la obra de forma virtual antes de llevarla a cabo, para poder predecir las dificultades y errores que se tendrán a la hora de realizarla, así como buscar las soluciones en caso de que se produzcan inconvenientes.

Por último, la interfaz gráfica deberá ser capaz de sincronizar, monitorizar y registrar en tiempo real los datos reales o simulados que recibe de ambos sistemas, con precisión y exactitud suficiente para poder guiar la máquina tuneladora.

A continuación, se presenta un esquema gráfico de las dimensiones reales de la máquina tuneladora modelo AVND 2000 AB. En este modelo se basa el desarrollo de la interfaz gráfica a la hora de tener en cuenta sus dimensiones. Véase figura 28.

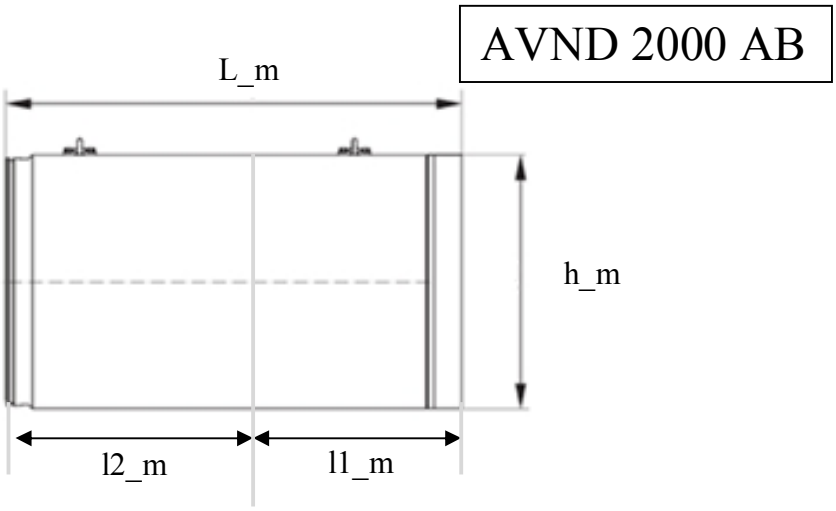


Figura 28. Dimensiones reales de la máquina tuneladora

	AVND 2000 AB
Longitud total [L_m]	3200 mm
Longitud de la parte delantera [l1_m]	1728 mm
Longitud de la parte trasera [12_m]	1472 mm
Diámetro exterior [h_m]	2425 mm

En la tabla anterior se pueden observar todas las magnitudes reales de ésta expresadas en milímetros con sus correspondientes nombres de variable.

Se ha elegido como modelo de referencia dentro del proyecto "La Ciudad Multidimensional" la máquina tuneladora AVND2000AB, ya que es el modelo más frecuente en obra. Véase Anexo A.

3.1.1 Requisitos del sistema

En general, una tuneladora es una máquina que permite perforar un túnel mediante un proceso continuo. En este proyecto se basará en el estudio para una tuneladora de tipo hinca de tubos. En este tipo de perforación, se utiliza el tubo como elemento definitivo del túnel y al mismo tiempo como elemento de empuje sobre la tuneladora. El avance se realiza gracias al empuje efectuado por un conjunto de cilindros de empuje instalados en el pozo de ataque sobre el tubo de hinca, el cual ha sido fabricado siguiendo unas normas estrictas, para poder soportar grandes esfuerzos longitudinales y transversales sin sufrir ningún deterioro. El tubo situado sobre el bastidor formara parte del túnel una vez concluida la hinca, cada tubo lleva instalada una junta en la boquilla, que debe garantizar la estanqueidad así como de

una "sufridera" en la cola, que absorbe las posibles irregularidades del tubo y que permite que este no sufra al unir dos tubos y empujar hormigón con hormigón. En todas las hincas se construye un pozo de ataque con un muro de reacción que soportara las presiones de empuje de toda la tubería y un pozo de llegada donde se rescatara la maquina. Para hincas de gran longitud, se instalan unos elementos entre los tubos que se denominan "Estaciones Intermedias", y que permiten distribuir los esfuerzos entre varios tramos. Véase figura 29.

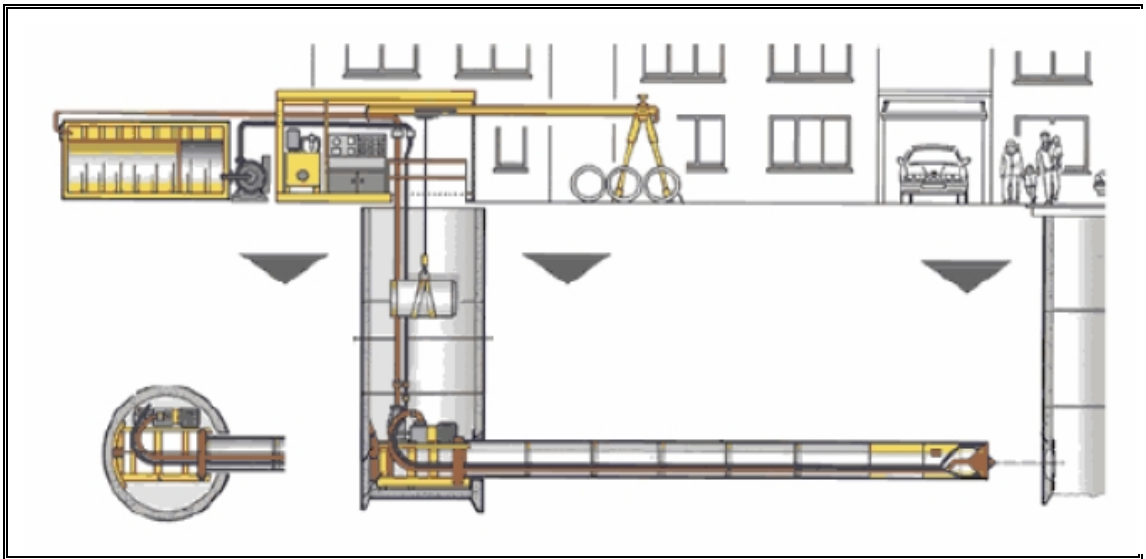


Figura 29. Diagrama de hinca de tubería en zona urbana

Hasta ahora, el guiado de una tuneladora, se realiza teleoperando desde una cabina de control, mediante la instrumentación de unos paneles de datos y mandos. El operario encargado del guiado, tiene la misión de regular el avance de la máquina dentro de unos límites óptimos prefijados.

Para realizar esta teleoperación, el operario necesita adquirir la experiencia suficiente para resolver cualquier situación que se produzca durante el uso de la máquina, adquiriendo una destreza en el manejo del control de los mandos, además de obtener suficiente comprensión de los posibles problemas que puedan ocurrir en el transcurso de las operaciones de perforación. En estos problemas, tales como desprendimientos excesivos de tierra, atascos en la extracción del material o aumentos de presión entre otros, debe ser capaz de tomar las decisiones oportunas para solventarlos o, en el peor de los casos, minimizar los riesgos que puedan surgir para las personas y la maquinaria.

A continuación, se muestra una imagen real del sistema de guiado completo que maneja un operario de máquina tuneladora. En el monitor de la parte derecha de la imagen, el operario tiene activado el simulador de hinca de tubos, donde configura toda la configuración de la máquina tuneladora para realizar el avance. Véase figura 30.

En la parte superior izquierda de la figura 30, se observa el monitor donde se encuentra cargada la interfaz gráfica externa comercial de la empresa alemana VMT, y el sistema de guiado SLS-Microtunnelling LT, véase referencia a su página web en bibliografía [8]. En esta pantalla, el operario monitoriza el avance de la máquina tuneladora mediante una flecha simbólica que tipifica la máquina tuneladora.

Este es el modelo práctico de guiado que se pretende alcanzar y mejorar su rendimiento mediante la implementación y sincronización de los nuevos componentes de guiado desarrollados.

Figura 30. Visualización del sistema de guiado completo

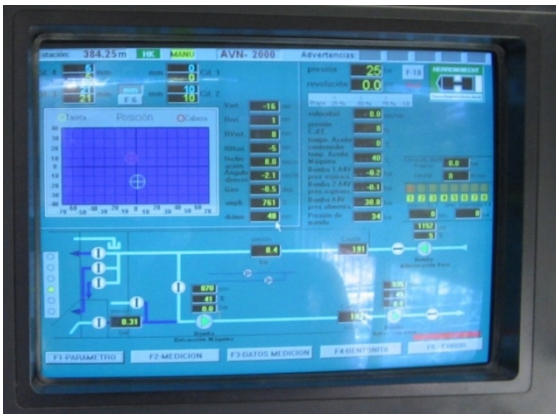


Figura 31. Visualización del panel de control del simulador de hinca de tubos

La labor primordial del operario es configurar los parámetros del simulador de hinca de tubos para conseguir que la flecha simbólica representada en la interfaz gráfica siga la trayectoria ideal esperada y realizar pequeñas modificaciones para corregir las desviaciones que se vayan produciendo.

Los componentes esenciales que forman el sistema de guiado mejorado que se pretende realizar son:

- Un teodolito o estación láser de puntería de color verde de 532nm. que tiene gran alcance y no le afectan en demasía las condiciones ambientales del túnel.
- Una diana con dos paneles de incisión, uno en la parte delantera y otro en la parte trasera de la diana, sobre los que incide el haz del láser.
- Un inclinómetro biaxial introducido en interior de la diana capaz de medir los ángulos que no se puedan calcular a partir de la incidencia del láser.
- Un ordenador con un software específico para el cálculo y procesamiento de los datos que definen la trayectoria de la máquina, a partir de los datos enviados por el resto de componentes, y que sea capaz de reproducirlos en pantalla, tanto de forma numérica como visual.

3.1.2 Requisitos funcionales

La interfaz gráfica de usuario debe ser un sistema fundamentalmente visual para que el operario que esté controlando la aplicación pueda tener en todo momento un conocimiento rápido y preciso de la posición y situación actual en que se encuentra la máquina tuneladora.

La información mostrada en la interfaz gráfica tiene que ser clara y concisa. Deben monitorizarse visiblemente tanto los valores numéricos más primordiales y transcendentes para el correcto guiado como las representaciones gráficas que indican visualmente la trayectoria o dirección que adopta la máquina tuneladora.

Se debe formar a los operarios mediante el manejo de la interfaz gráfica en modo simulación que utilice técnicas de realidad virtual, al estilo de los utilizados en otras ramas de la industria (aeronáutica, transporte férreo). Este tipo de plataformas permitirán una formación del operario en un entorno casi real y ofrecen la posibilidad de simular cualquier situación que pueda darse en la práctica.

Las comunicaciones entre los diferentes módulos implicados en el sistema de guiado deben estar correctamente sincronizadas para realizar automáticamente todo el flujo de intercambios de variables sin producirse ningún fallo de lectura o escritura de datos.

El tiempo de ejecución de cada software debe de estar totalmente depurado para que las variables almacenadas en el fichero de intercambio de datos lleguen rápidamente a la interfaz gráfica externa para que ésta pueda monitorizar los avances producidos por la máquina tuneladora prácticamente en tiempo real.

3.1.3 Sistemas de referencia

En este apartado se detallan los sistemas de referencia que se han puesto en común entre todos los bloques que integran el sistema de guiado, para cumplir la característica comentada anteriormente de unificar criterios a la hora de obtener todos los parámetros necesarios.

El sistema de referencia global común para todos los módulos, se ha emplazado al pozo de ataque del túnel, eligiendo como posición de origen el centro de la sección transversal del túnel. En cuanto a los ejes, se ha tomado el eje Z en la dirección de avance de la máquina, cumpliendo que junto al resto formen un sistema dextrógiro.

Para los ángulos, se ha llegado al acuerdo de utilizar como referencia los ángulos de Euler, siendo el ángulo alfa el de "pitch" o cabeceo de la máquina, el ángulo beta el de "yaw" o guiñada, y gamma el de "roll" o alabeo. Se reproduce este sistema, para su mejor comprensión. Véase figura 32.

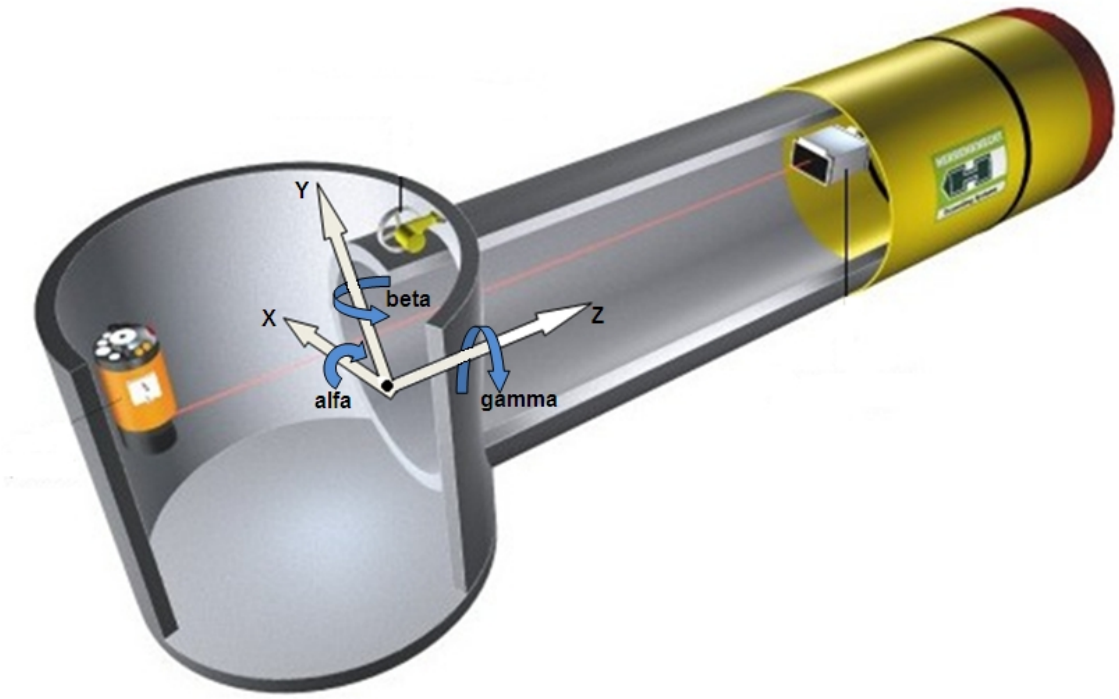


Figura 32. Sistema de referencia global de la máquina tuneladora

Para la diana, además, se llevan dos sistemas de referencia auxiliares, los cuales están ubicados en el centro de los dos paneles con los que cuenta, y con los ejes paralelos a los del sistema de referencia global. Con ellos se calculan los puntos de incidencia del láser para, posteriormente, calcular los ángulos y desviaciones realizados por la máquina tuneladora. Por último, se referencia al sistema global el punto ubicado bajo el panel trasero de la diana ortogonal al eje de simetría de la máquina, como punto de cálculo de la trayectoria.

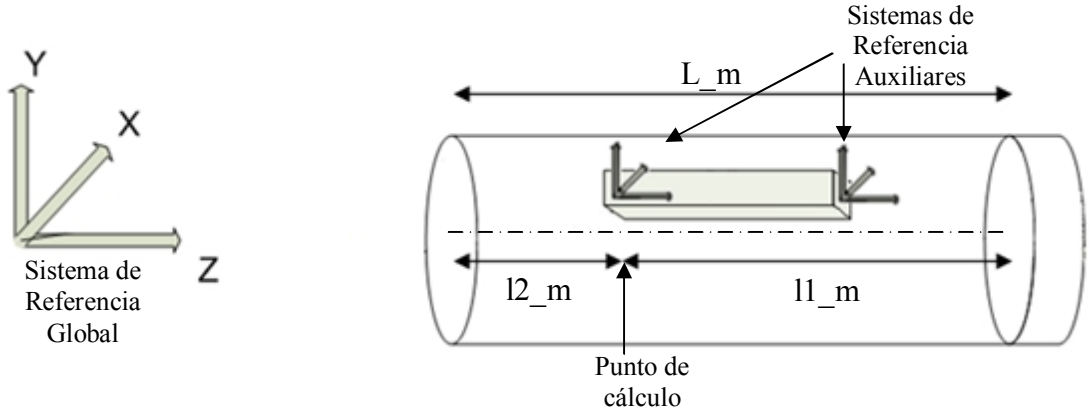


Figura 33. Sistemas de referencia de la máquina tuneladora y de la diana

El simulador de hinca de tubos tiene una referencia al mismo punto de cálculo que la diana, y mantiene el sistema de referencia global, por lo que es equivalente a la imagen anterior. Véase figura 33.

Para finalizar, la interfaz gráfica de usuario GUI, transforma el punto de la trayectoria obtenido tanto por la diana como por el simulador, a los puntos trasero y delantero de la máquina, respecto del sistema de referencia global y apoyándose en un sistema de referencia móvil ubicado bajo la parte trasera de la diana, a la altura del eje de simetría de la máquina tuneladora. Véase figura 34.

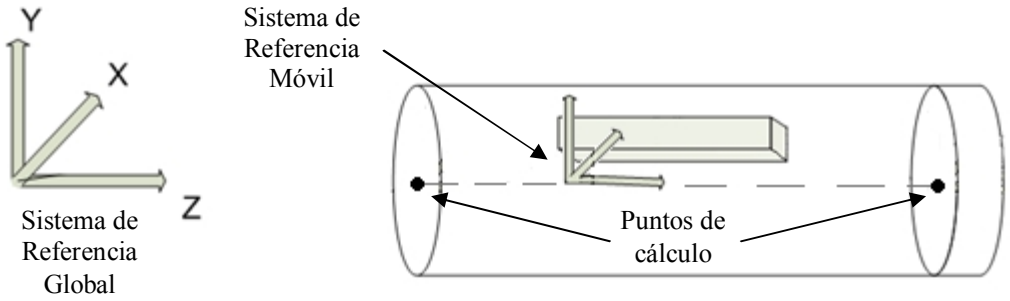


Figura 34. Puntos de cálculo de coordenadas de la parte delantera y posterior de la máquina tuneladora

Sin embargo, aunque todos los cálculos para mostrar y registrar la información numérica se realizan como se ha indicado, no es así para el caso de los puntos en la representación visual, puesto que debido a las grandes dimensiones de la máquina tuneladora con respecto al pequeño avance de la misma, cada vez que se muestra una nueva representación, se verían solapadas las flechas que simbolizan la trayectoria de la tuneladora.

Con el fin de resolver este problema, se ha realizado un escalado a la hora de representar la tuneladora gráficamente, según se muestra en la imagen a continuación. Véase figura 35.

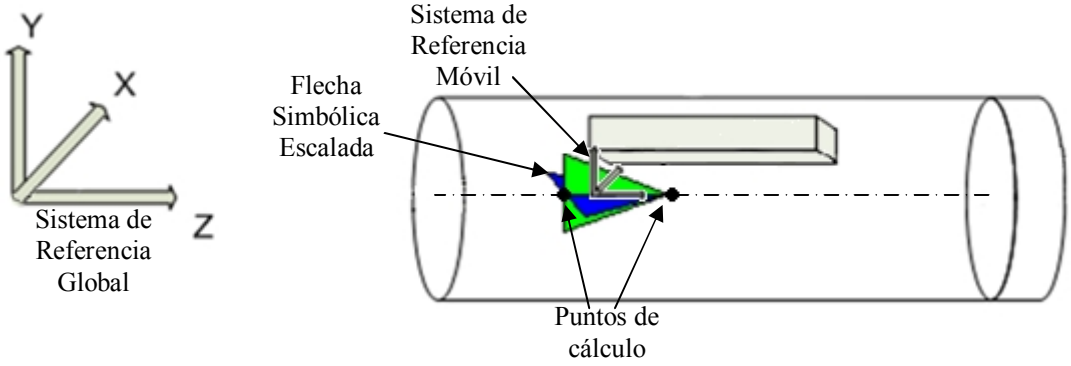


Figura 35. Puntos de cálculo de la parte delantera y trasera de la flecha simbólica

3.2 Arquitectura del sistema de guiado

El sistema de guiado de la máquina tuneladora AVND 2000 AB, cuenta con tres módulos bien diferenciados encargados de obtener, procesar, representar y transmitir los datos necesarios para realizar un guiado lo más exacto y preciso a la trayectoria ideal marcada por los topógrafos de la obra.

En primer lugar se encuentra la diana inteligente, encargada de obtener todas las coordenadas reales de posición y ángulos de giro durante cada avance de la máquina tuneladora. La única coordenada de posición real que la diana no puede calcular es el punto kilométrico de la distancia recorrida desde el inicio de la obra. Esta coordenada de posición es generada por el simulador de hinca de tubos y transmitida a través del módulo de la GUI. A partir de que la diana inteligente recibe dicho dato de avance, generará el resto de coordenadas reales que serán transmitidas al módulo GUI para la realización de los cálculos previos a la representación gráfica y paramétrica.

En segundo lugar se encuentra el simulador de hinca de tubos, encargado de obtener todas las coordenadas simuladas de posición y ángulos de giro que se realizan durante cada avance de la máquina tuneladora. Estas coordenadas son transmitidas y almacenadas en el módulo GUI para su posterior representación gráfica y paramétrica.

En tercer lugar se encuentra el módulo GUI, encargado de coordinar todo el flujo de datos. Recibe todas las coordenadas del módulo de la diana inteligente y transmite los parámetros calculados por la GUI al simulador de hinca de tubos. El intercambio de información por parte de los tres módulos debe de estar rigurosamente sistematizado para que el sistema de guiado no se colapse por la gran cantidad de datos entrantes y salientes durante todo el recorrido físico de la máquina tuneladora.

Otra función esencial del módulo GUI es la representación gráfica tanto en dos dimensiones como en tres dimensiones de datos reales y simulados. Del mismo modo, debe quedar representada y almacenada toda la información paramétrica que necesita el operario para el correcto guiado de la máquina tuneladora. El módulo GUI ofrece al operario la opción de representación real o simulada y además poder alternar y comparar entre ambas.

3.2.1 Interfase: Simulador-GUI-Diana

En este apartado se va a explicar detalladamente el flujo de operación y datos que mantiene el simulador de hinca de tubos y la diana inteligente respecto al módulo de la GUI. Véase figura 36.

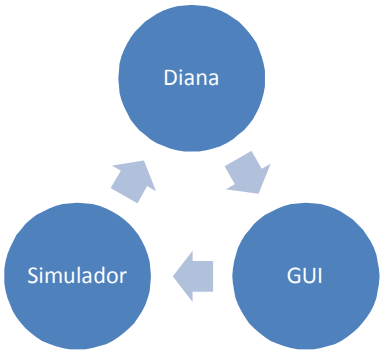


Figura 36. Flujo de operación Simulador-Diana-GUI

En primer lugar partimos de la base en que los tres módulos se encuentran inicializados y preparados para el comienzo de la tunelación.

La diana inteligente antes de comenzar a generar coordenadas de posicionamiento (x, y) y ángulos de giro (alfa, beta, gamma) con el avance de la máquina, necesita recibir desde el simulador de hinca de tubos el parámetro del punto kilométrico de avance (z). A partir de este momento, la diana calculará y enviará al módulo GUI el resto de coordenadas (x, y, alfa, beta, gamma) teniendo en cuenta dicha variable (z).

Los avances kilométricos (z) provenientes del simulador de hinca de tubos serán cada muy corto espacio de tiempo pero el cálculo de coordenadas reales dependerá del tiempo que tarde en realizar las operaciones de cálculo el software de la diana.

Por tanto, con las coordenadas recibidas de la diana y el punto kilométrico de avance (z) del simulador, el módulo GUI podrá realizar sus cálculos y representaciones a partir de estas variables.

La información enviada al módulo GUI por parte de la diana será con un intervalo de tiempo lo suficientemente pronunciado para que el operario pueda observar los cambios que vayan sucediendo a lo largo del proceso de tunelación.

El simulador, como se ha explicado anteriormente, está continuamente generando todas las coordenadas de posición (x_simulada, y_simulada, z_simulada) y ángulos de giro (alfa_simulada, beta_simulada, gamma_simulada) con un alto índice de repetitividad. Esta información se envía al módulo GUI que mantiene un control del flujo de datos para poder realizar las representaciones, tanto paramétricas como gráficas, con un intervalo de tiempo adecuado para que el operario pueda observar los cambios que vayan sucediendo a lo largo del proceso de tunelación.

El módulo GUI trasforma las coordenadas reales recibidas de la diana al sistema de referencia global de la máquina tuneladora para poder trasladar las coordenadas a otros sistemas de referencia más útiles y trabajar con más facilidad. Estos sistemas de referencia son la parte delantera y la parte trasera de la máquina tuneladora.

Estas nuevas coordenadas trasformadas son enviadas al simulador de hinca de tubos para que represente en su diana gráfica del panel de mandos la parte delantera y trasera de la máquina con el objetivo de llevar un seguimiento de su trayectoria y sus tendencias.

Las coordenadas enviadas son: "x_cabeza, y_cabeza, x_posterior, y_posterior, angulo_direccion, inclinación".

Sin embargo, cada vez que se produce una representación de coordenadas simuladas por parte del módulo GUI, las coordenadas generadas simuladas no son enviadas al simulador de hinca de tubos.

3.2.2 Procesado de información

En este apartado, se va a desarrollar la transición de flujo de la información de datos, procesados desde la obtención de los parámetros iniciales de referencia, hasta la visualización final de la posición espacial de la máquina tuneladora.

Como ya se ha comentado anteriormente, para el sistema de guiado se necesitan tres bloques fundamentales, compuestos de una diana láser, un simulador y una interfaz gráfica externa (GUI).

El sistema de guiado puede adquirir tanto datos reales procedentes de la diana, como datos procedentes del simulador. La interfaz gráfica debe ser capaz de procesar toda la información recibida por medio de la diana o del simulador para su respectiva monitorización.

Por este motivo, se establece un sistema de referencia global para todos los módulos que componen el sistema de guiado, con el fin de poder unificar un criterio común a la hora de especificar las coordenadas posicionales en las que se ubica la máquina tuneladora. Este sistema de referencia global está situado en la entrada del túnel y referenciado al eje de simetría de la máquina tuneladora.

La diana láser trabaja bajo dos sistemas de referencia auxiliares ubicados en el centro de ambos paneles de incidencia, donde recoge la posición de los puntos generados por el láser, a partir de la cual calcula los ángulos y las desviaciones producidas en la máquina para calcular la nueva situación espacial de la tuneladora. Estas coordenadas son transformadas al sistema de referencia global como salida del sistema hacia la interfaz gráfica.

El simulador genera tanto coordenadas de posición, como desviaciones angulares, a partir de la configuración de los parámetros del sistema empuje y cabezal de corte, que contralan la trayectoria ficticia que sigue la máquina.

El simulador ha adoptado el mismo punto de referencia que la diana para obtener la trayectoria simulada con respecto al sistema de referencia global.

La interfaz gráfica recibe las coordenadas de la trayectoria, tanto reales como simuladas, y las transforma a las nuevas posiciones correspondientes de la parte delantera y trasera de la máquina tuneladora.

Gracias a la unificación de criterios, tanto de los sistemas de referencia como del punto de cálculo por parte de la diana y el simulador, los cálculos realizados para la transformación de coordenadas son iguales para ambos módulos.

La monitorización paramétrica de los valores calculados para la obtención de las posiciones y desviaciones angulares de la parte delantera y trasera de la máquina tuneladora, corresponden con las dimensiones reales de ésta.

Sin embargo, para la monitorización gráfica, tanto en dos como en tres dimensiones, se realiza un escalado reductor de las dimensiones totales de ésta. Se representa una flecha simbólica que tipifica la máquina tuneladora, para conseguir representar un fragmento ampliado que permita visualizar con más precisión los cambios de trayectoria.

Del mismo modo, se realiza un escalamiento aumentativo de la desviación angular del eje de avance longitudinal para visualizar exaltadamente los cambios angulares que resultan inapreciables.

A continuación, se va explicar detalladamente las variables con las que trabaja cada uno de los tres módulos y que comparten entre ellos.

En primer lugar, la diana láser calcula las variables respecto a los dos sistemas de referencia auxiliares, “x1”, y1”, “x2” e “y2”, que son las desviaciones de los puntos producidos por la incidencia del láser en los paneles delantero y trasero de la diana. A partir de los mismos, se calculan las variables finales “x”, “y”, “alfa”, “beta” y “gamma” respecto al sistema de referencia global. La unidad elegida para las variables de posición es el milímetro, mientras que las variables angulares se calculan en radianes.

En segundo lugar, el simulador de hinca de tubos realiza el cálculo de variables rigiéndose en una librería dinámica, programada en C++, que obtiene las variables simuladas de posición “x_simulada”, “y_simulada”, “z_simulada”, angulares “alfa_simulada”, “beta_simulada”, “gamma_simulada”, y parámetros del láser “amplitud” y “diametro”. La unidad elegida para las variables de posición es el milímetro, mientras que las variables angulares se calculan en grados, y los parámetros del láser en tanto por ciento y milímetros respectivamente.

En tercer lugar, la interfaz gráfica calcula los puntos situados en la parte trasera y delantera de la máquina tuneladora, a partir tanto de los puntos reales, obtenidos por la diana, como de los ofrecidos por el simulador. Además, para realizar una comprobación en el caso de estar usando el sistema de diana real y simulador al mismo tiempo, hará entrega de los valores reales calculados para los puntos de la parte trasera y delantera de la máquina al simulador.

Como variables de salida hacia los otros módulos por parte de la GUI, tenemos:

Las variables correspondientes a los valores reales de la posición de la máquina son "x_head_real", "y_head_real", "x_button_real" e "y_button_real". La unidad utilizada para el cálculo de parámetros correspondientes a las dimensiones de la máquina tuneladora es el milímetro.

Las variables correspondientes al cálculo de tendencias son "inclinacion" y "angulo_direccion". La unidad utilizada para el cálculo de de las tendencias es milímetro partido de metro.

A continuación, en la figura 37 se expone el diagrama de bloques de los tres módulos explicados anteriormente con sus respectivas interacciones y flujo de datos correspondientes. Además se muestran los nombres de las variables utilizadas.

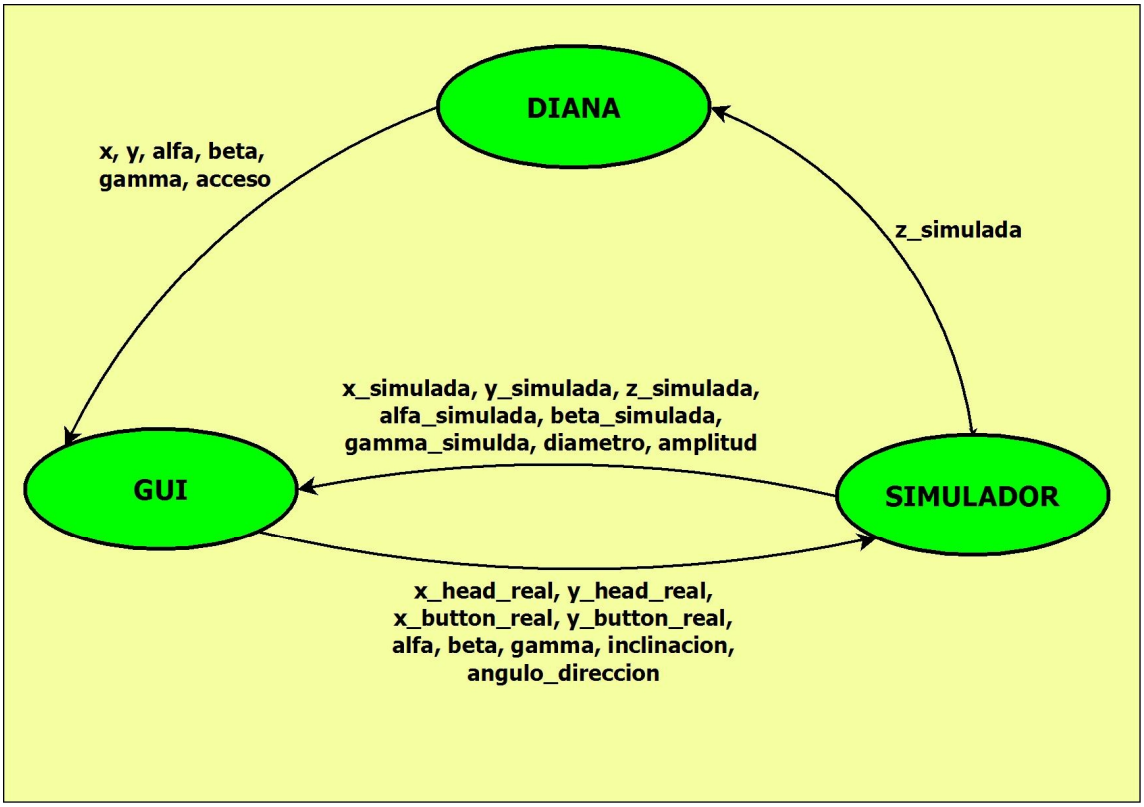


Figura 37. Intercambio de variables entre los tres módulos que forman el sistema de guiado

3.3 Entorno de programación MATLAB

3.3.1 Entorno gráfico y creación de interfaces

Hoy en día, con la proliferación de los sistemas operativos tipo Windows, en los que el usuario encuentra un ambiente amigable y fácil de usar, donde se puede ingresar datos en cajas de texto y ejecutar códigos con sólo hacer clic en un botón, MATLAB no podía quedarse atrás.

MATLAB es un gran programa de cálculo técnico y científico. Para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización. En otras aplicaciones resulta bastante más lento que el código equivalente desarrollado en C/C++ o Fortran. Sin embargo, siempre es una magnífica herramienta de alto nivel para desarrollar aplicaciones técnicas, fácil de utilizar y que aumente significativamente la productividad de los programadores respecto a otros entornos de desarrollo.

Las últimas versiones de MATLAB se han preocupado por incorporar herramientas fáciles de usar, con las que el usuario pueda crear interfaces gráficas para la interacción con sus programas.

MATLAB posee la herramienta GUIDE (Graphical Use Interface Development Environment) que permiten crear interfaces gráficas y tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++ pero con muchas menos posibilidades. Estas son las llamadas GUI (Guide User Interface) y podemos acceder a ellas tecleando la instrucción “guide” en la ventana de comandos.

Encontraremos doce herramientas con las que podemos crear nuestra interfaz: Push Button, Toggle Button, Radio Button, CheckBox, Edit Text, Static Text, Slider, Table, Panel, ListBox, Popup Menu, Axes. Así mismo, encontraremos un formulario en blanco sobre el cual podemos empezar a construir el nuestro proyecto.

En general, familiarizarse a la programación de una interfaz gráfica de MATLAB es un poco más complicado que programarla en otros programas de creación de interfaces gráficas. Por ello se puede usar el comando “Help”, que en sus últimas versiones tiene guías paso a paso para crear GUI’s.

Sin embargo, el entorno de diseño y trabajo de MATLAB es muy gráfico, intuitivo y fácil de manejar.

3.3.2 Características

Algunas de las características principales que hacen del MATLAB un programa intuitivo y de gran utilidad técnica se exponen a continuación:

- Computación numérica para obtener rápidamente resultados precisos.
- Gráficos para visualizar y analizar los datos.
- Lenguaje y entorno de programación interactivos.
- Herramientas para construir GUI's a medida.
- Integración con aplicaciones externas como C, C++, Fortran, Java, componentes COM o Excel.
- Posibilidad de importar datos desde archivos y dispositivos externos y de usar archivos E/S de bajo nivel (además de acceso a bases de datos y hardware adicional a través de productos añadidos).
- Conversión de aplicaciones MATLAB a C y C++ mediante su compilador.

Este amplio conjunto de prestaciones hacen de MATLAB la base ideal para el desarrollo de soluciones a problemas matemáticos, creación de interfaces gráficas y comunicaciones con otros programas.

3.3.3 Entorno flexible de trabajo

El entorno de trabajo de MATLAB es muy gráfico e intuitivo, similar al de otras aplicaciones profesionales de *Windows*.

Está diseñado para la computación interactiva y automatizada. Mediante las funciones matemáticas y gráficas incorporadas y las herramientas de fácil manejo se pueden analizar y visualizar los datos de un vistazo. El lenguaje estructurado y las herramientas de programación permiten guardar los resultados de las exploraciones interactivas y desarrollar algoritmos y aplicaciones.

Los usuarios que trabajan con una amplia gama de aplicaciones de diversos niveles de complejidad han encontrado en MATLAB un entorno eficaz y flexible que evoluciona a su mismo ritmo.

Con MATLAB se puede experimentar de forma interactiva y probar ideas propias con sólo aplicar la gama de funciones de análisis y gráficos del producto. El escritorio de MATLAB permite acceder fácilmente a sus archivos de datos, programas, gráficos, ayuda en línea del producto y mucho más. También puede configurarse para que contenga las herramientas con las que se trabaja más a menudo.

A través de la interfaz del escritorio de MATLAB, se puede personalizar para adaptarlo a cualquier estilo de trabajo y usar selectivamente las funciones que sean necesarias en cada fase del proyecto.

Los componentes más importantes del entorno de trabajo de MATLAB son:

1. El Escritorio de MATLAB (*MATLAB Desktop*)

   El *MATLAB Desktop* es la ventana más general de la aplicación. El resto de las ventanas o componentes pueden alojarse en el *MATLAB Desktop* o ejecutarse como ventanas independientes.

   A su vez, los componentes alojados en el *MATLAB Desktop* pueden aparecer como sub-ventanas independientes o como pestañas dentro de una de las sub-ventanas.

   MATLAB ofrece una gran flexibilidad al respecto y es cada usuario quien decide en qué forma desea utilizar la aplicación.

   La figura 38 muestra el menú *Desktop*, desde el que se controlan los componentes visibles y la forma en que se visualizan. La configuración adoptada por el usuario se mantendrá la siguiente vez que arranque el programa.

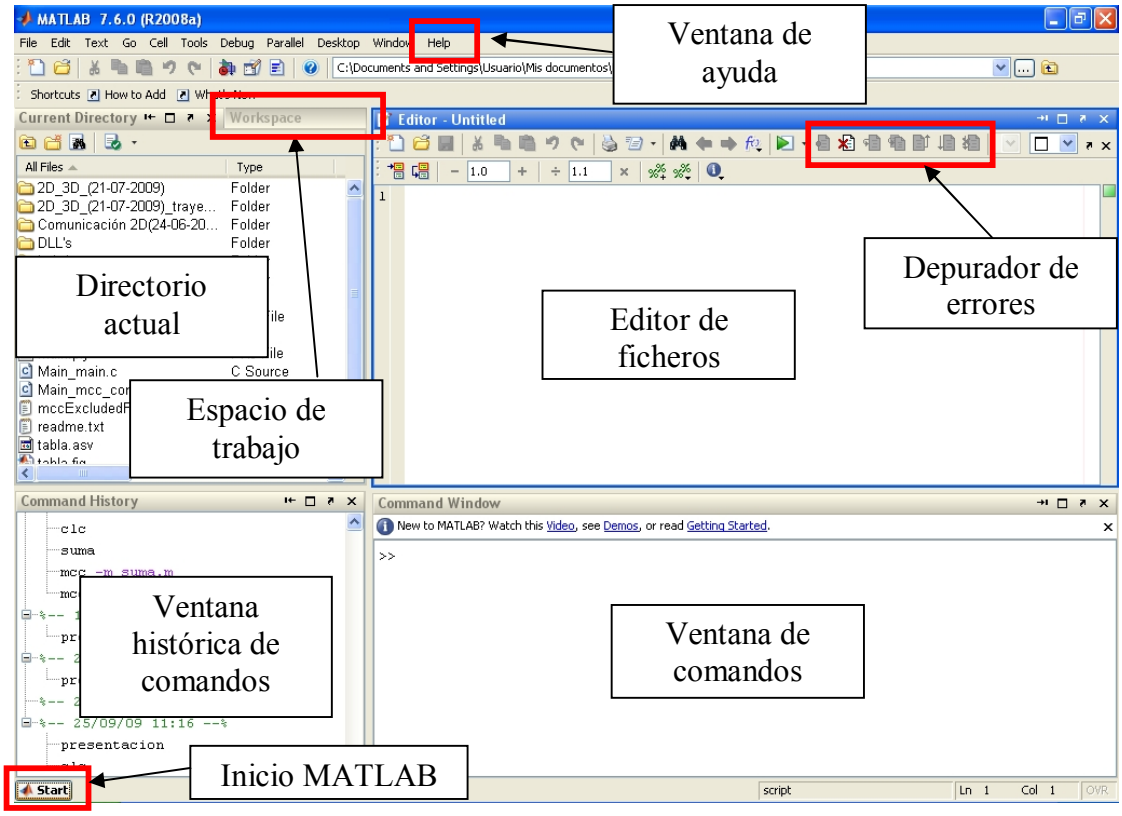


Figura 38. Vista del escritorio de MATLAB (MATLAB Desktop)

2. **Los componentes individuales orientados a tareas concretas**:
   a) La ventana de comandos (*Command Window*).
   b) La ventana histórica de comandos (*Command History*).
   c) El directorio actual (*Current Directory*).
   d) El espacio de trabajo (*Workspace*).
   e) El editor de ficheros y depurador de errores (*Editor&Debugger*).
   f) La ventana de ayuda (*Help*).

En la parte inferior izquierda de la pantalla aparece el botón *Start*, con una función análoga a la del botón *Inicio* de *Windows*. *Start* da acceso inmediato a los módulos o componentes de MATLAB que se tengan instalados.

A continuación se describen brevemente estos componentes. Utilizar MATLAB y desarrollar programas para MATLAB es mucho más fácil si se conoce bien este entorno de trabajo para alcanzar la máxima productividad personal en el uso de esta aplicación.

a) La ventana de comandos (*Command Window*)

Ésta es la ventana en la que se ejecutan interactivamente las instrucciones de MATLAB y en donde se muestran los resultados correspondientes si se ejecutan operaciones matemáticas. Es la ventana más importante y la única que existía en las primeras versiones de la aplicación.

b) La ventana histórica de comandos (*Command History*)

La ventana *Command History* ofrece acceso a las sentencias que se han ejecutado anteriormente en la *Command Window*. Estas sentencias están también accesibles por medio de las teclas ↑ y ↓ como en las versiones anteriores, pero esta ventana facilita mucho el tener una visión más general de lo hecho anteriormente y seleccionar lo que realmente se desea repetir.
Las sentencias ejecutadas anteriormente se pueden volver a ejecutar mediante un doble clic o por medio del menú contextual que se abre al clicar sobre ellas con el botón derecho. También se pueden copiar y volcar sobre la línea de comandos, pero se ha de copiar toda la línea, sin que se admita la copia de un fragmento de la sentencia. Existen opciones para borrar algunas o todas las líneas de esta ventana.

c) El directorio actual (*Current Directory*)

La ventana *Current Directory* permite explorar los directorios del ordenador en forma análoga a la del *Explorador* u otras aplicaciones de *Windows*. Cuando se llega al directorio deseado se muestran los ficheros allí contenidos. La ventana *Current Directory* permite ordenarlos por fecha, tamaño, nombre, etc. El directorio actual cambia automáticamente en función del directorio seleccionado con este explorador, y también se puede cambiar desde la propia barra de herramientas del *Matlab Desktop*.

Para que un fichero **.m* se pueda ejecutar es necesario que se cumpla una de las dos condiciones siguientes:

1. Que esté en el *directorio actual*. Este directorio es el primer sitio en el que MATLAB busca cuando desde la línea de comandos se le pide que ejecute un fichero.

2. Que esté en uno de los directorios indicados en el *Path* de MATLAB. El *Path* es una lista de directorios donde el programa busca los ficheros que ha de ejecutar. Muchos de los directorios del *Path* son propios de MATLAB, pero los usuarios pueden añadir sus propios directorios.

d) El espacio de trabajo (*Workspace*)

El espacio de trabajo de MATLAB (*Workspace*) es el conjunto de variables y de funciones de usuario que en un determinado momento están definidas en la memoria del programa o de la función que se está ejecutando.

Éstas son las variables del espacio de trabajo base, es decir, el de la línea de comandos de MATLAB. Cada función tiene su propio espacio de trabajo, con variables cuyos nombres no interfieren con las variables de los otros espacios de trabajo.

La ventana *Workspace* constituye un entorno gráfico para ver las variables definidas en el espacio de trabajo. La figura 34 muestra el aspecto inicial de la ventana *Workspace* cuando se abre desde un determinado programa.

Hay que tener en cuenta que cuando se termina de ejecutar una función y se devuelve el control al programa que la había llamado, las variables definidas en la función dejan de existir salvo que se hayan declarado persistentes y también deja de existir su espacio de trabajo.

e) El editor de ficheros y depurador de errores (*Editor&Debugger*)

En MATLAB tienen particular importancia los ficheros *m-files*. Son ficheros de texto ASCII, con la extensión **.m*, que contienen conjuntos de comandos o definición de funciones. La importancia de estos ficheros **.m* es que al teclear su nombre en la línea de comandos y pulsar *Intro*, se ejecutan uno tras otro todos los comandos contenidos en dicho fichero. El poder guardar instrucciones y grandes matrices en un fichero permite ahorrar mucho trabajo de tecleado.

Aunque los ficheros **.m* se pueden crear con cualquier editor de ficheros ASCII tal como *Notepad*, MATLAB dispone de un editor que permite tanto crear y modificar estos ficheros, como ejecutarlos paso a paso para ver si contienen errores, este es el proceso de *Debug* o depuración.

El *Editor* muestra con diferentes colores los diferentes tipos o elementos constitutivos de los comandos, en verde los comentarios, en violeta las cadenas de caracteres, etc. El *Editor* se preocupa también de que las comillas o paréntesis que se abren, no se queden sin el correspondiente elemento de cierre. Seleccionando varias líneas y clicando con el botón derecho aparece un menú contextual cuya sentencia "Comment" permite entre otras cosas comentar con el carácter *%* todas las líneas seleccionadas.

Estos comentarios pueden volver a su condición de código ejecutable seleccionándolos y ejecutando "Uncomment" en el menú contextual.

Otra opción muy útil de ese menú contextual es "Smart Indent", que organiza el sangrado de los bucles y bifurcaciones de las sentencias seleccionadas.

La ejecución del *Debugger* comienza eligiendo el comando *Run* en el menú *Debug*, pulsando la tecla F5, clicando en el botón *Continue* () de la barra de herramientas del *Editor* o tecleando el nombre del fichero en la línea de comandos de la *Command Window*.

Los puntos rojos que aparecen en el margen izquierdo son "breakpoints", es decir, puntos en los que se detiene la ejecución de programa. La flecha verde en el borde izquierdo indica la sentencia en que está detenida la ejecución antes de ejecutar dicha sentencia.

Cuando el cursor se coloca sobre una variable aparece una pequeña ventana con los valores numéricos de esa variable.

A continuación, se detalla el significado de los botones pertenecientes al depurador de errores:

 **Set/Clear Breakpoint**. Coloca o borra un "breakpoint" en la línea en que está el cursor.

 **Clear All Breakpoints**. Elimina todos los "breakpoints" que haya en el fichero.

 **Step**. Avanzar un paso sin entrar en las funciones de usuario llamadas en esa línea.

 **Step In**. Avanzar un paso, y si en ese paso hay una llamada a una función cuyo fichero **.m* está accesible, entra en dicha función.

 **Step Out**. Salir de la función que se está ejecutando en ese momento.

 **Continue**. Continuar la ejecución hasta el siguiente "breakpoint".

 **Quit Debugging**. Terminar la ejecución del *Debugger*.

El *Debugger* es muy útil para detectar y corregir errores y además sirve para aprender métodos numéricos y técnicas de programación.

f) La ventana de ayuda (*Help*)

MATLAB dispone de un excelente menú de ayuda (*Help*) con el que se puede encontrar la información que se desee. La figura 39 muestra las distintas opciones que aparecen en el menú *Help* de la ventana principal.

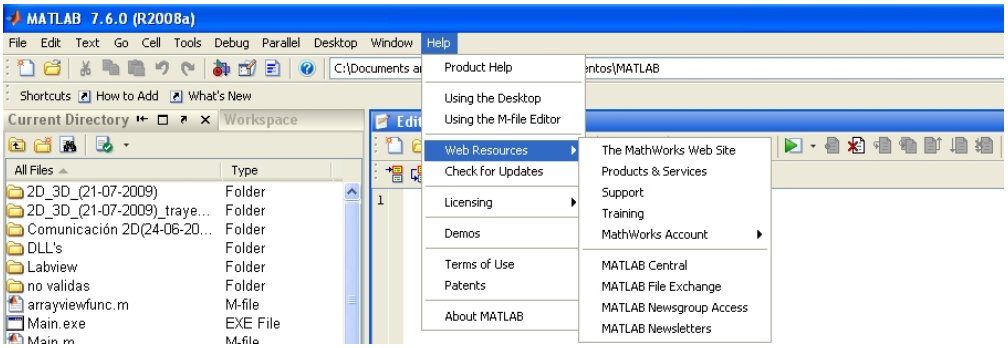


Figura 39. Algunas páginas web sobre MATLAB

En resumen, MATLAB dispone de una ayuda muy completa y accesible, estructurada en varios niveles (línea de comandos en la *Command Window*, ventana *Help*, y manuales en formato PDF), con la que es muy importante estar familiarizado, porque hasta los más expertos programadores tienen que acudir a ella con una cierta frecuencia.

3.3.4 Diseño y configuración de la GUI

Para iniciar la herramienta GUIDE que se va a utilizar a la hora de realizar el proyecto, se pueda hacer de dos maneras:

- Ejecutando la siguiente instrucción en la ventana de comandos:

```
>> guide
```

- Haciendo un clic en el icono que muestra la figura 40.

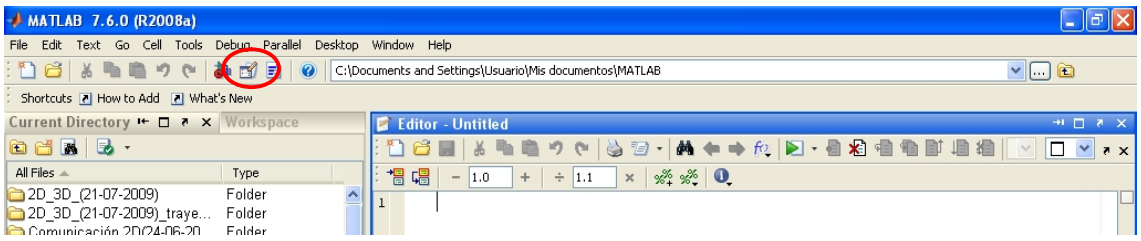


Figura 40. Icono de inicio de la herramienta GUIDE

Se presentan las siguientes opciones:

a) GUI en blanco (Por defecto)

La opción de interfaz gráfica de usuario en blanco (viene predeterminada), nos presenta un formulario nuevo, en el cual podemos diseñar nuestro programa.

b) GUI con Controles

Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades. Podemos ejecutar este ejemplo y obtener resultados.

c) GUI con Ejes y Menú

Esta opción es otro ejemplo el cual contiene el menú *File* con las opciones *Open*, *Print* y *Close*. En el formulario tiene un *Popup menu*, un *push button* y un objeto *Axes*, podemos ejecutar el programa eligiendo alguna de las seis opciones que se encuentran en el menú despegable y haciendo clic en el botón de comando.

d) Pregunta de diálogo Modal

Con esta opción se muestra en la pantalla un cuadro de diálogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones "Yes" y "No", dependiendo del botón que se presione, el GUI retorna el texto seleccionado (la cadena de caracteres 'Yes' o 'No').

Elegimos la primera opción, GUI en blanco. Véase figura 41.

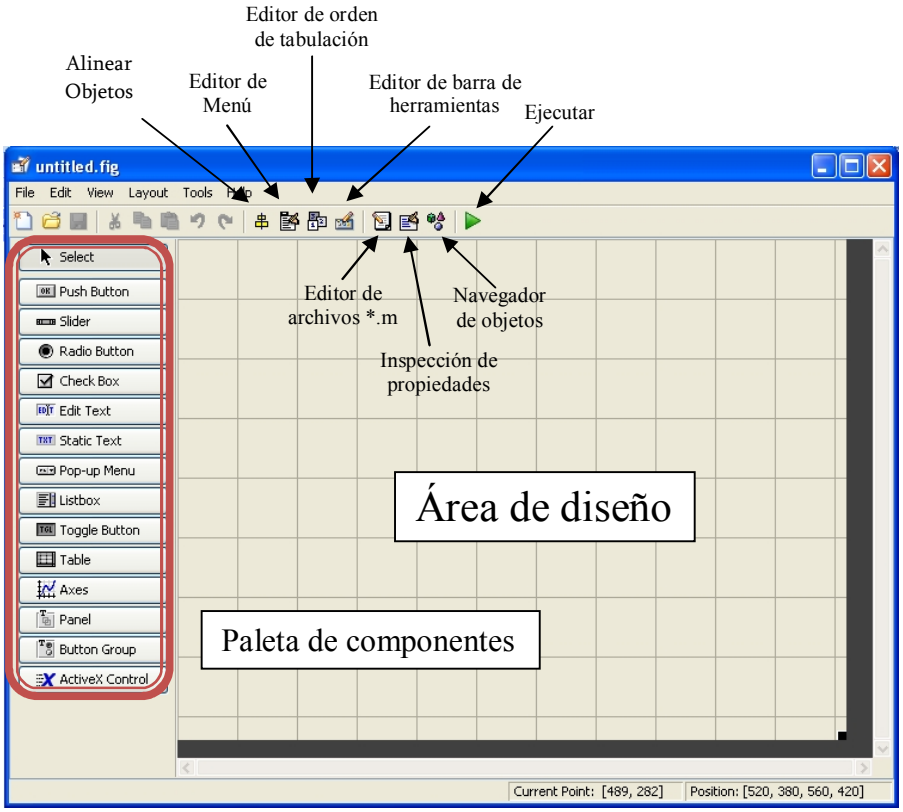


Figura 41. Entorno de diseño: componentes etiquetados

A continuación, se va explicar el proceso de diseño de la interfaz gráfica externa y cómo se ha ido desarrollando y utilizando todas las herramientas de la paleta y propiedades de configuración para cada elemento.

1. Se ha coloreado el fondo de la interfaz gráfica utilizando el menú "Property Inspector" de la barra de herramientas y seleccionando el color deseado en el menú contextual de las propiedades de la figura. Además, se ha activado la opción "ToolBar" para disponer de la paleta de herramientas gráficas a la hora de ejecutar la interfaz gráfica externa. Véase figura 42.

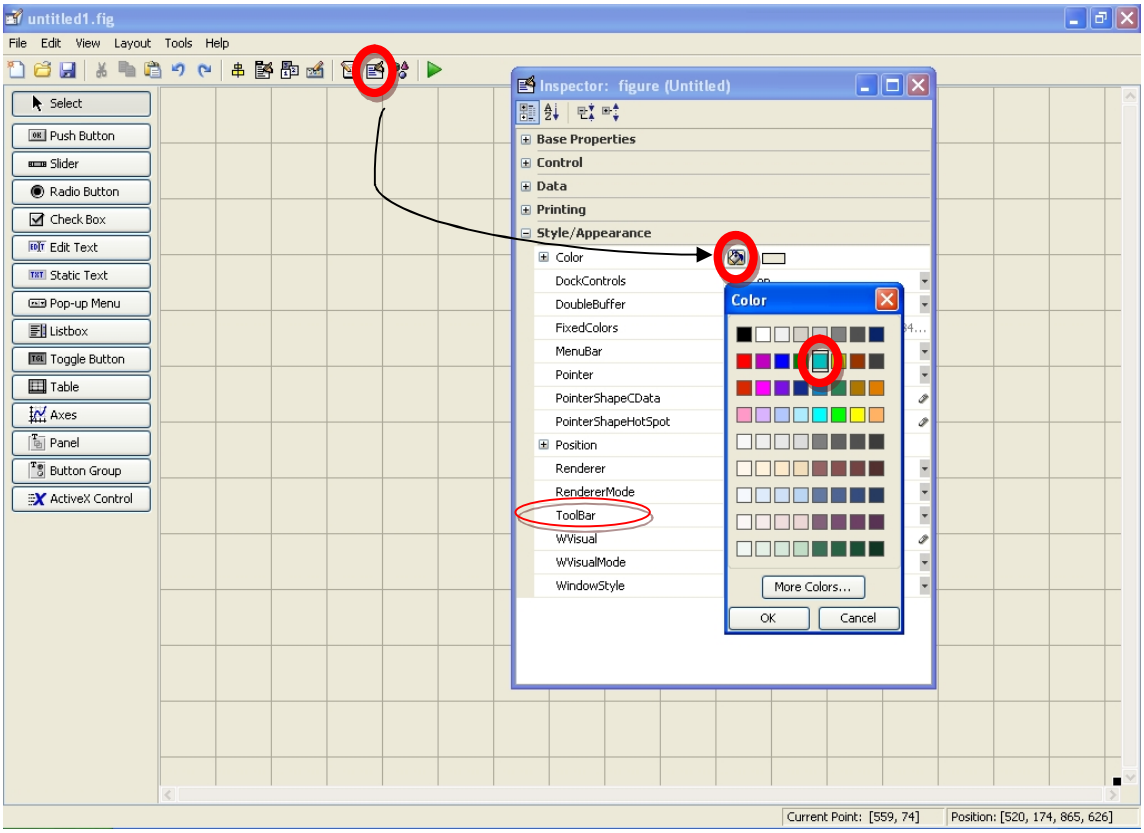


Figura 42. Configuración del color del fondo de pantalla

2. Se han añadido todos los paneles que forman el diseño de la interfaz gráfica y se han configurado mediante la opción "Property Inspector" de la barra de herramientas para pintarlos del color deseado, incrementar su relieve y escribir el título o encabezado de cada panel en caso de que sea necesario.

   Un panel tampoco no es un control propiamente dicho. Su función es la de englobar una serie de opciones (botones, cajas de texto, etc.…) con el fin de mantener una estructura ordenada de controles, separando unos de otros en función de las características del programa y del gusto del programador.

A continuación, se visualiza el orden y distribución de todos los paneles creados en la interfaz gráfica y las opciones seleccionadas para su configuración. El nombre del panel seleccionado se configura mediante la opción "Title" del menú contextual de propiedades y para el relieve se selecciona la opción "Border Type" y se elige el tipo "beveledout". Véase figura 43.

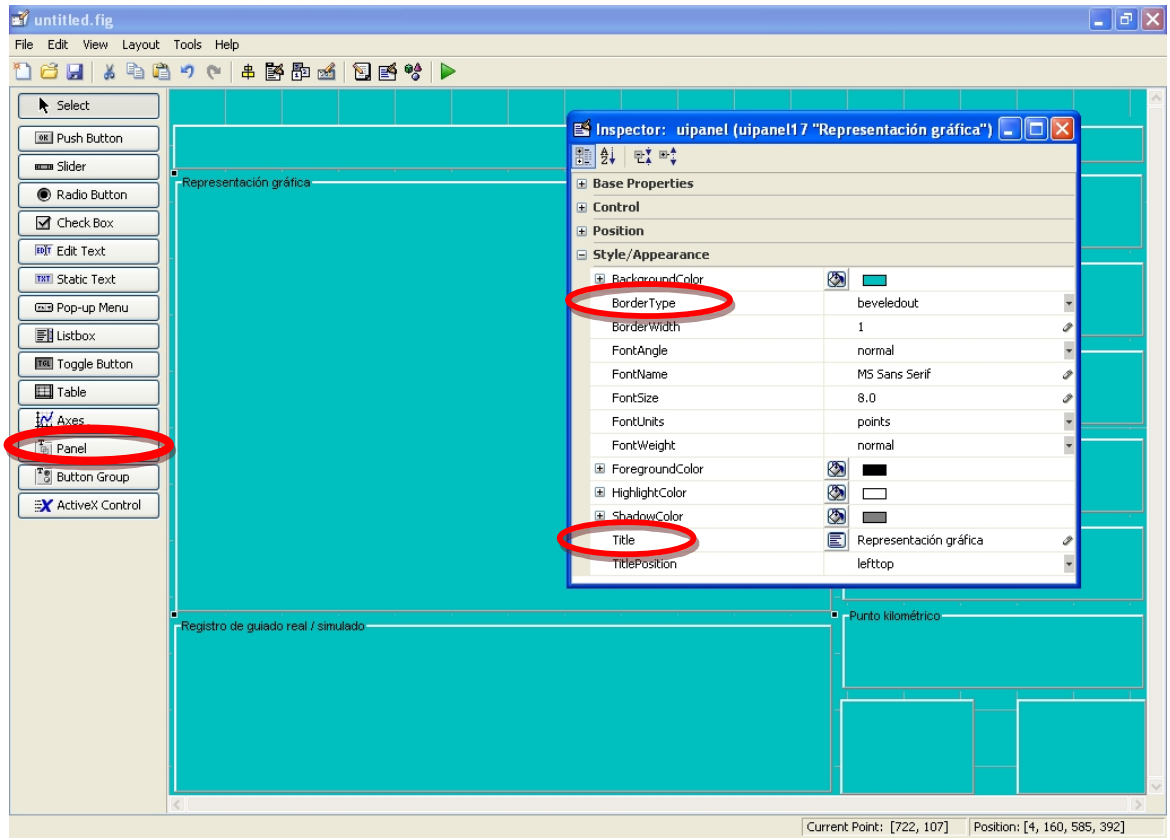


Figura 43. Configuración del título y de los paneles

Para poder realizar una alineación y distribución homogénea de todos los paneles que se han creado en la interfaz, se seleccionan los que se desean ordenar y se pulsa el icono "Align Objects" en la paleta de herramientas, desplegándose un menú contextual con variedad de opciones a elegir. Véase figura 44.

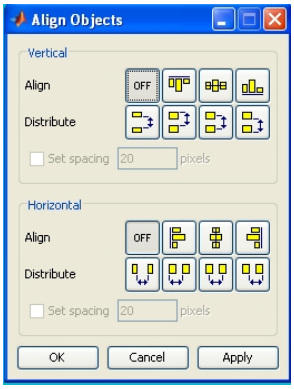


Figura 44. Alineación de los componente de la interfaz gráfica

3. Se incluyen todos los textos estáticos de los paneles de control y del título de la interfaz gráfica. Cada nombre de variable, cada indicador numérico y cada unidad de magnitud es un texto estático independiente correctamente alineado mediante la opción anteriormente explicada "Align Objects". Véase figura 45.

Los textos estáticos no son controles, ya que no permite realizar ninguna operación con el ratón. Permiten escribir un texto en una caja en la pantalla.

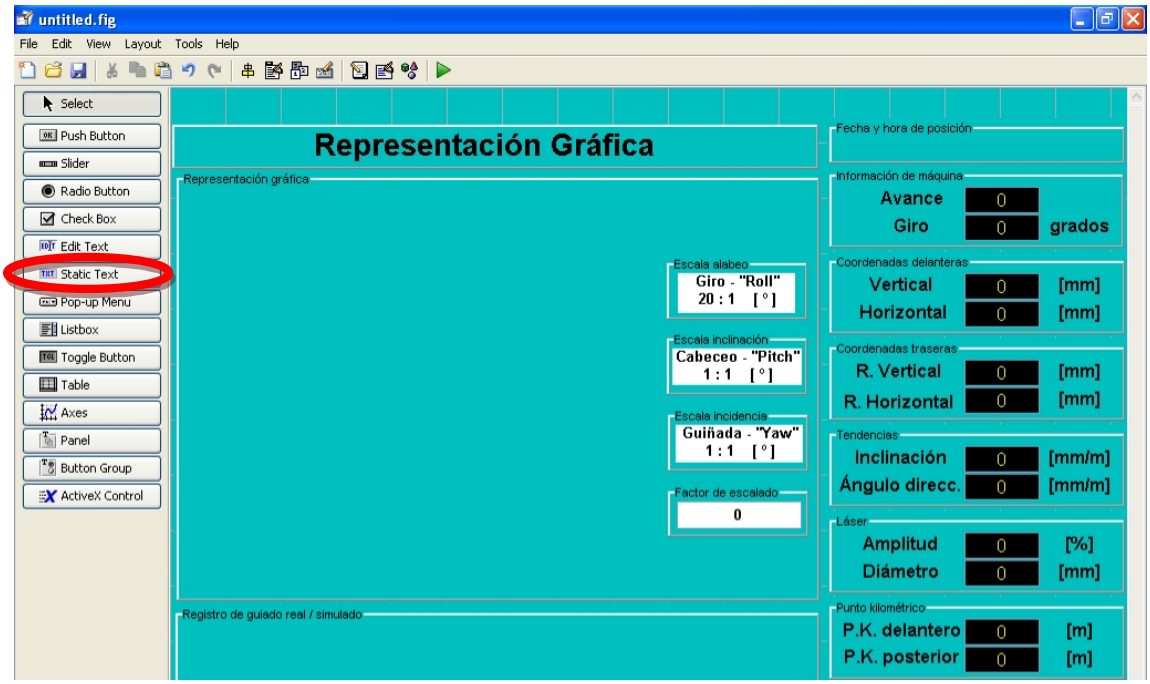


Figura 45. Colocación de los textos estáticos

Para realizar el diseño gráfico de los indicadores numéricos, se configuran sus opciones de formato seleccionando el indicador en cuestión y pulsando el icono de la paleta de herramientas "Property Inspector". Véase figura 46.

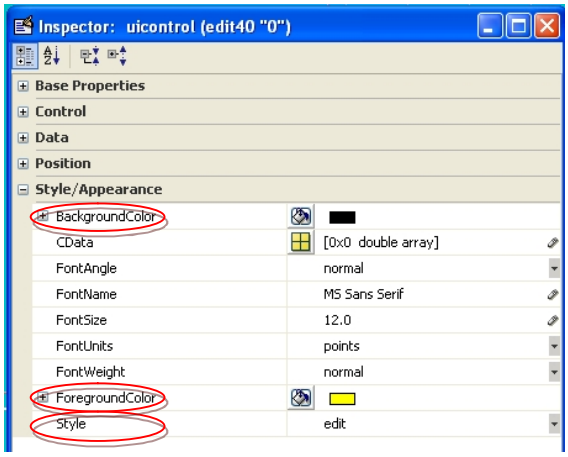
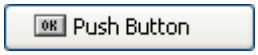


Figura 46. Configuración del texto estático

Se configura el color de fondo negro del texto estático mediante la opción "BackgroundColor" y el color amarillo de los números a través de la opción "ForegroundColor". Por último, se cambia el estilo mediante la opción "Style" y se elige el tipo "edit" para ahondar el indicador numérico y dar más sensación de realismo.

4. Se introducen mediante el componente de la paleta "Push Button" los cuatro botones de interacción de la interfaz gráfica.



Los botones son pequeños objetos de la pantalla normalmente acompañados con texto. Al pulsar sobre ellos con el ratón genera una acción y su rutina de llamada se ejecuta. Véase figura 47.

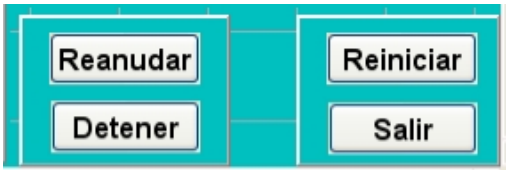


Figura 47. Botones de la interfaz gráfica

Se configura el formato del texto del interior del botón del mismo modo que para los componentes anteriores y se cambia el nombre del botón mediante la opción "String". Véase figura 48.

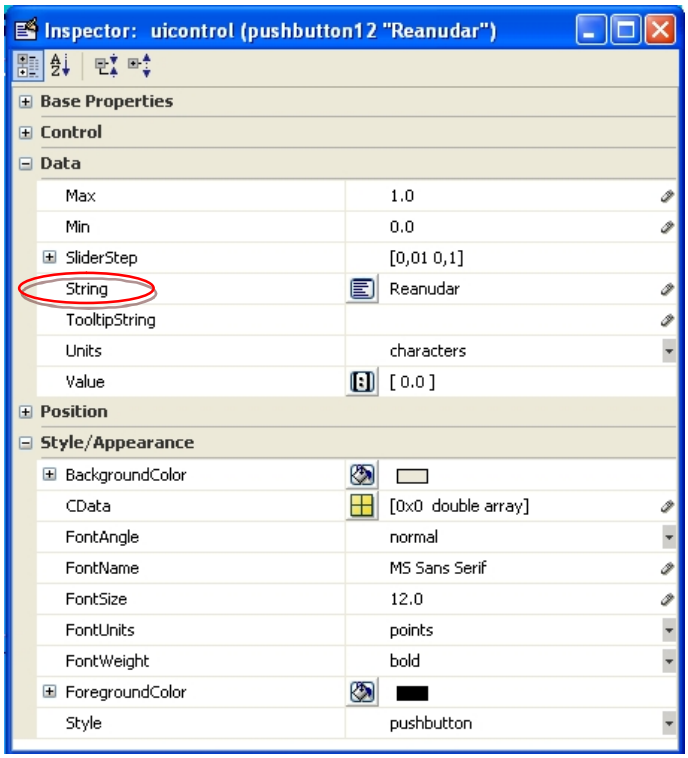


Figura 48. Configuración de los botones de la interfaz gráfica

5. Se integra en la interfaz gráfica la tabla de registro de guiado pulsando la opción "Table" en la paleta de componentes.

El componente "Table" permite la creación de una tabla de valores, donde se pueden leer y almacenar datos establecidos por el usuario. Véase figura 49.

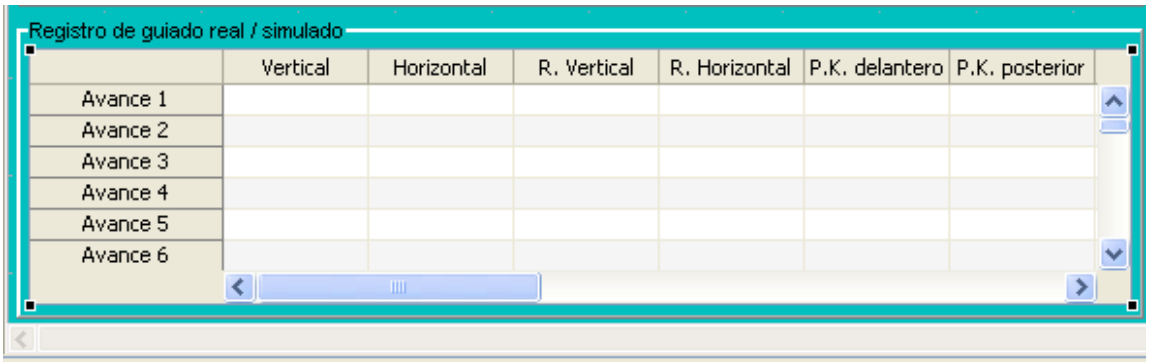


Figura 49. Tabla de registros de guiado

Se configura la tabla de registros pulsando sobre ella con botón derecho del ratón y eligiendo la opción "Table Property Editor" en el menú contextual generado. Véase figura 50.

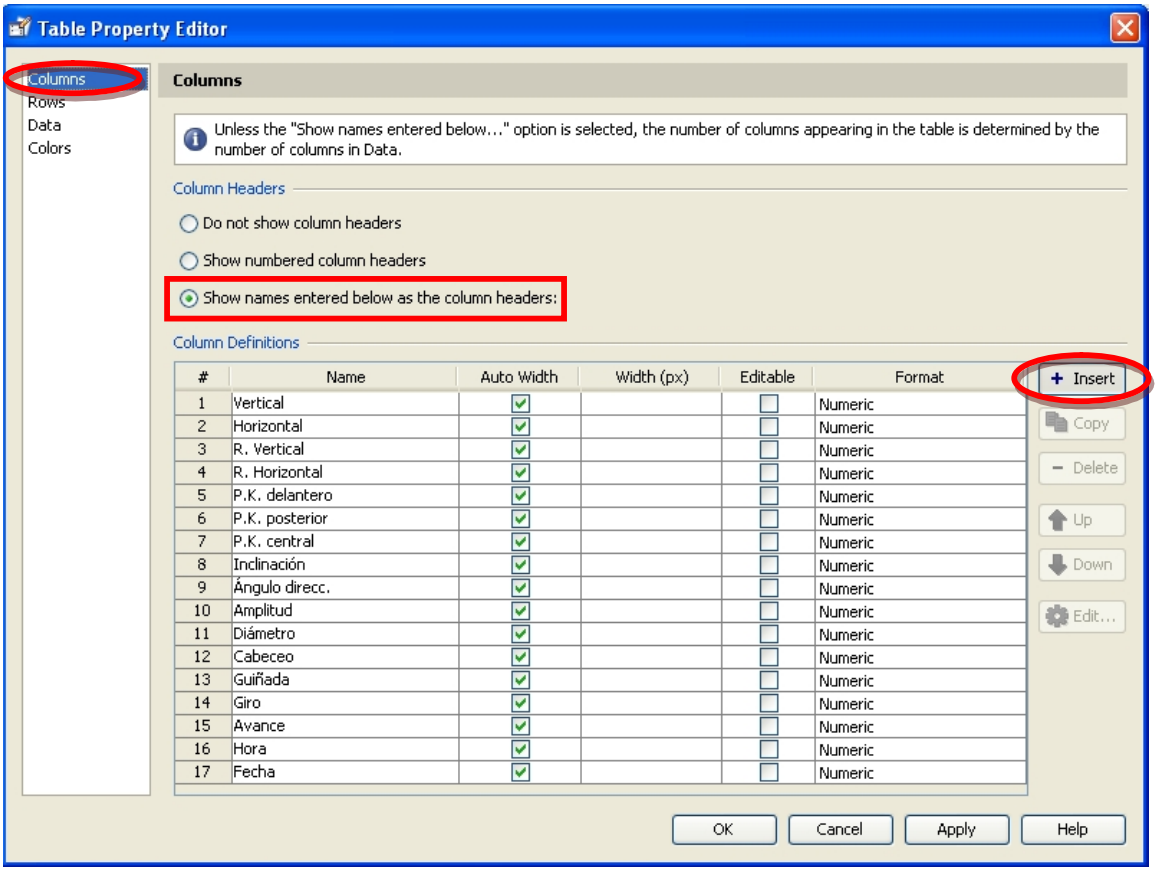
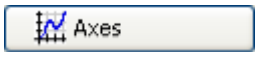


Figura 50. Configuración de la tabla de registros de guiado

Para configurar el número de columnas, se selecciona la opción "Columns" en el editor de propiedades de la tabla y se pulsa el botón "Insert" para insertar en número de columnas deseado. Se marca la opción "Show names entered below as the column headers" para que se represente el nombre introducido de cada columna a la hora de la representación gráfica y se elige formato "Numeric" ya que se va a trabajar con parámetros numéricos.

Del mismo modo, para la configuración de las filas, se selecciona en el editor de propiedades de la tabla la opción "Rows" y se sigue el mismo procedimiento que en el caso anterior.

6. Se introduce en la interfaz los ejes para la representación gráfica pulsando en la paleta de componentes la opción "Axes".



Crea un área donde se pueden hacer representaciones gráficas de coordenadas tanto en dos dimensiones como en tres dimensiones, se puede añadir una cuadrícula, configurar la escala numérica y escribir etiquetas de los ejes.

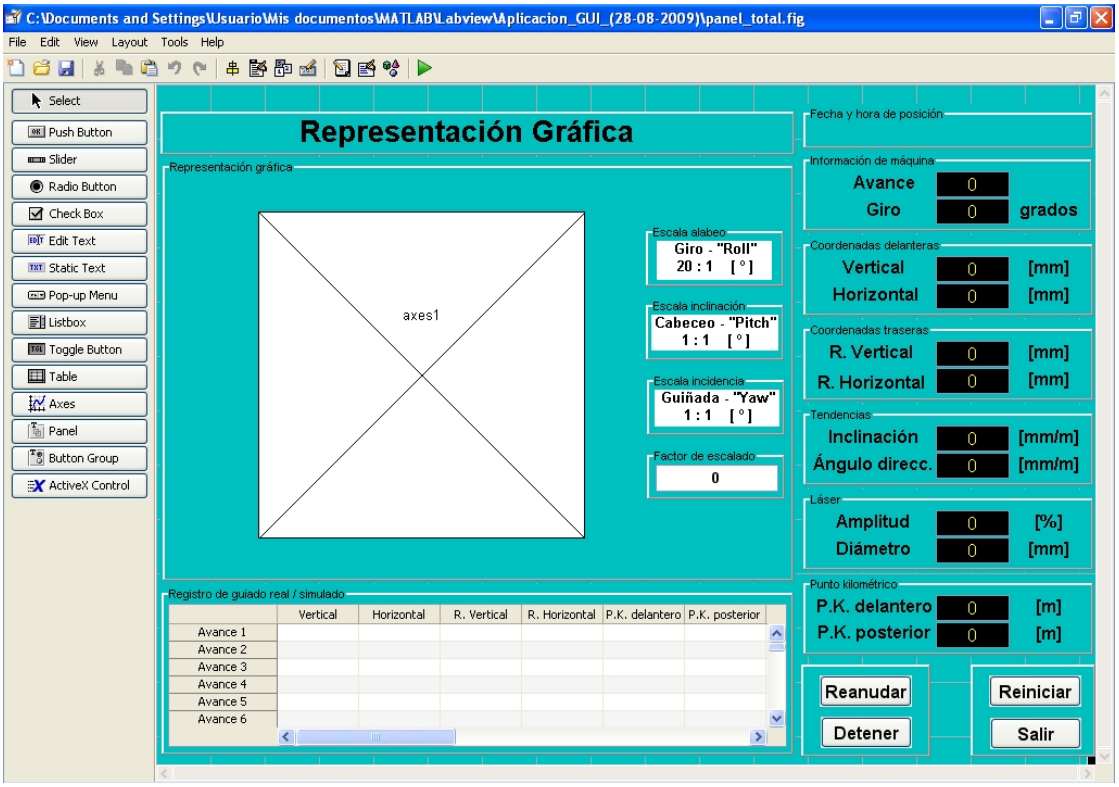


Figura 51. Interfaz gráfica de usuario

Para realizar la configuración de las propiedades de los ejes, se selecciona el componente ejes y se pulsa sobre el icono "Property Inspector" para desplegar un menú contextual con todas las opciones tanto gráficas como paramétricas de los tres ejes (X, Y, Z).

A continuación, se muestra el menú de propiedades disponible para el componente "Axis". Véase figura 52.

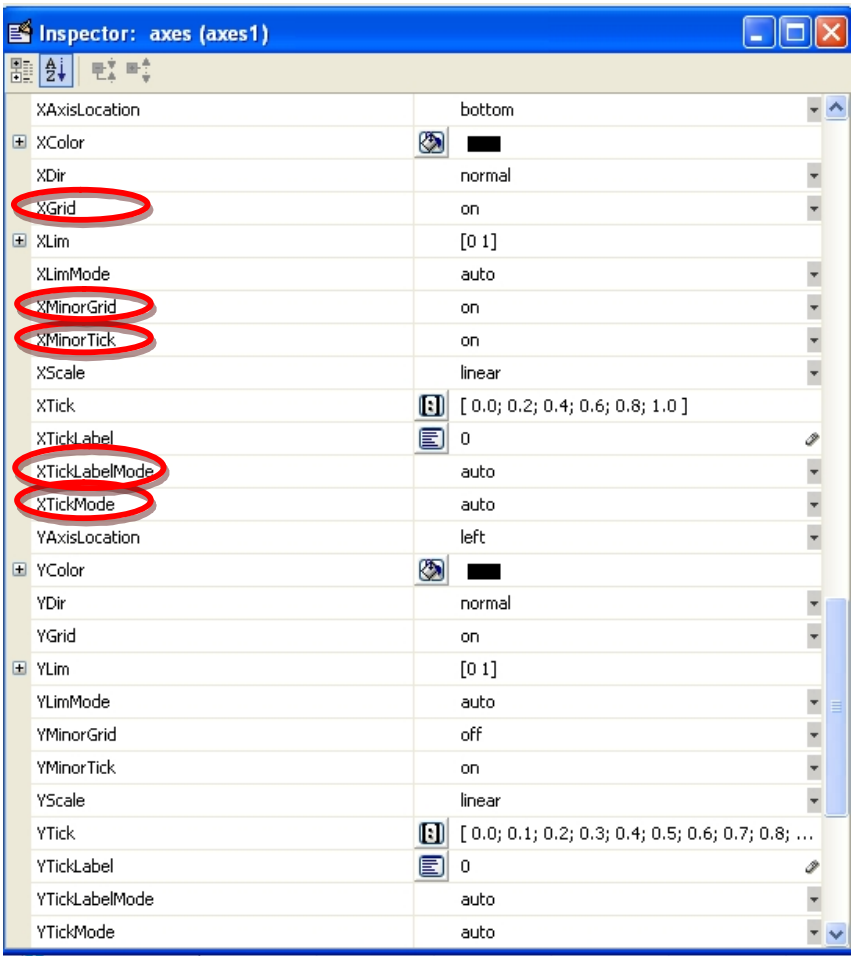


Figura 52. Configuración del eje de coordenadas de la interfaz gráfica

Algunas de las propiedades más importantes que se deben configurar para optimizar la representación gráfica de las coordenadas son las que respectan a la escala de los ejes y a su punto de vista.

- "XGrid" → Dibuja una cuadrícula en el eje X.
- "XMinorGrid" → Dibuja una cuadrícula mucho más habitada en el eje X.
- "XLimMode" → Permite alternar entre modo automático y manual para introducir los límites superior e inferior en el eje X.
- "XTickLabelMode" → Permite alternar entro modo automático y manual para introducir las etiquetas de numeración del eje X.
- "XTickMode" → Permite alternar entre modo automático y manual para introducir la escala de numeración del eje X.

7. Se insertan los menús interactivos en la interfaz gráfica mediante el icono de la barra de herramientas "Menu Editor". Esta herramienta permite crear varios menús de forma jerarquizada. Véase figura 53.

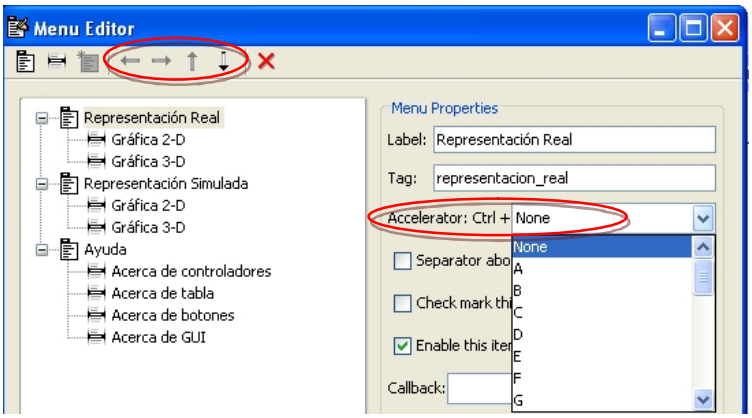


Figura 53. Configuración de los menús de la interfaz gráfica

Mediante las flechitas colocadas en la parte superior de la figura anterior, se puede mover el menú seleccionado a un nivel superior o inferior. Además se le puede añadir una tecla de acceso rápido configurando la opción "Accelerator".

8. Por último, se guarda los cambios realizados en la interfaz gráfica y se ejecuta pulsando el icono "Run" de la barra de herramientas.

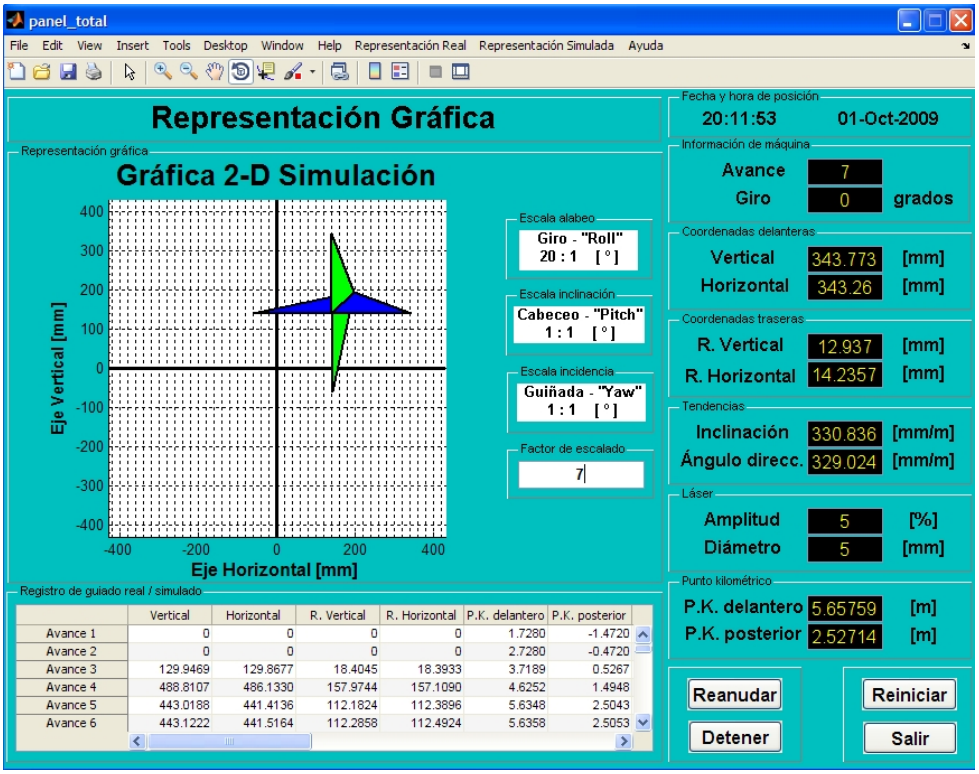


Figura 54. Presentación de la GUI en ejecución

En la figura anterior, se muestra la interfaz de guiado después de diseñar y configurar todos los componentes y de lanzar su ejecución. Véase figura 54.

3.3.5 Funcionamiento de una aplicación GUI

Una aplicación GUIDE consta de dos archivos: **.m* y **.fig*.

- El archivo **.m* es el que contiene el código con las correspondencias de los elementos de control de la interfaz gráfica. Cada vez que se introduzca un elemento gráfico en el panel de diseño de la interfaz (**.fig)* se generará unas líneas de programa automáticamente asociadas a ese tipo de control. Estas líneas de programas están vacías, es decir, es necesarios programar acciones a elementos de control para llevarlas a cabo durante la ejecución del programa.

- El archivo **.fig* es el que contiene los elementos gráficos así como las propiedades de la interfaz. Cada vez que se adicione un nuevo elemento en la interfaz gráfica, se genera automáticamente código en el archivo **.m*.

Para ejecutar una interfaz gráfica, si la hemos etiquetado con el nombre de *ejemplo.fig*, simplemente se ejecuta en la ventana de comandos

>> *ejemplo*

O bien, haciendo clic derecho en el **.m* correspondiente y seleccionando la opción "Run". Véase figura 55.

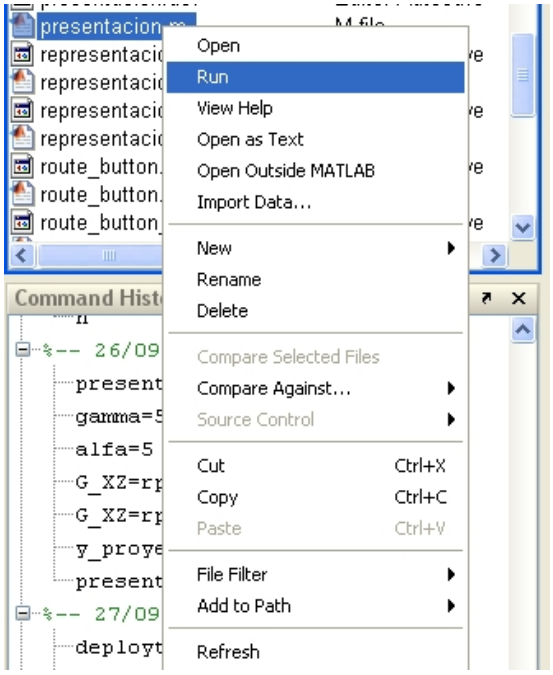


Figura 55. Ejecutar una GUI desde el directorio de trabajo

3.3.6 Flujo de operación

El concepto básico de la operación del software con una interfaz gráfica de usuario es que el flujo de cómputo está controlado por las diferentes acciones en la interfaz.

Mientras que en un guión el flujo de comandos está predeterminado, el flujo de operaciones con una GUI no lo está. Los comandos para crear una interfaz con el usuario se escribe en un guión, la interfaz invoca el guión que se ejecute, mientras la interfaz del usuario permanece en la pantalla aunque no se haya completado la ejecución del guión.

Cuando se interactúa con un control, el programa registra el valor de esa opción y ejecuta los comandos prescritos en la cadena de invocación.

El control guarda un *string* que describe la acción a realizar y cuando se invoca puede consistir en un solo comando de MATLAB, en una secuencia de comandos o en una llamada a una función. Es recomendable utilizar llamadas a funciones, sobre todo cuando se requieren de más de unos cuantos comandos en la invocación.

Los menús de interfaz con el usuario, los botones, los menús desplegables, los controladores deslizantes y el texto editable son dispositivos que controlan las operaciones del software. Al completarse la ejecución de las instrucciones de la cadena de invocación, el control vuelve a la interfaz para que puedan elegirse otra opción del menú. Este ciclo se repite hasta que se cierra la interfaz gráfica.

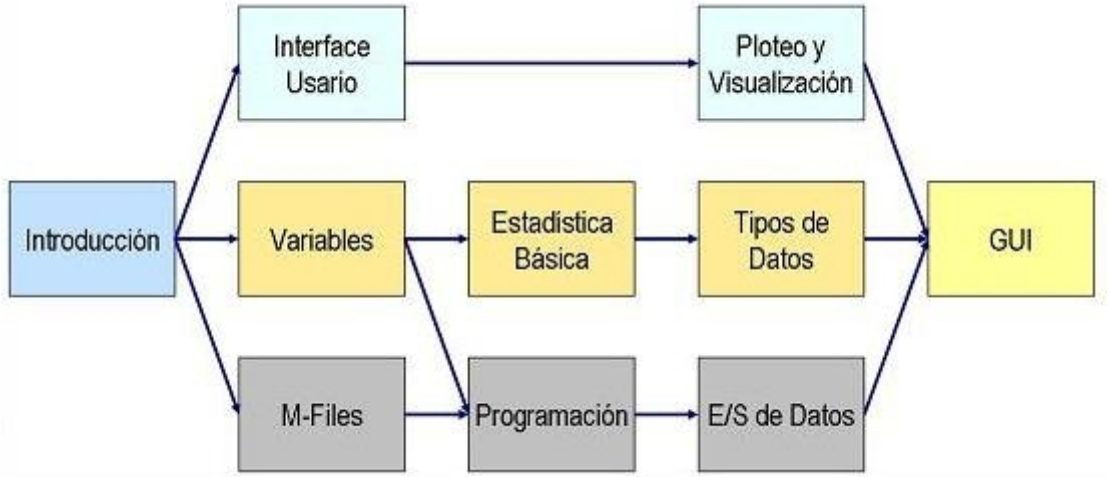


Figura 56. Ciclo de operación de la GUI

En la figura anterior, se muestra un esquema gráfico del flujo de operación que hay que tener en cuenta a hora de diseñar y desarrollar una interfaz gráfica externa. Véase figura 56.

3.4 Implementación del software de la interfaz gráfica

3.4.1 Descripción del sistema

El sistema de la interfaz gráfica de usuario (GUI) está formado por treinta archivos o funciones **.m* y varios archivos de intercambio de variables **.mat*. Todos estos archivos se encuentran almacenados dentro de una carpeta de trabajo del programa MATLAB. Véase listado de funciones en Anexo B.

El sistema de monitorización del guiado de la máquina tuneladora comienza cuando se ejecuta la función "presentacion.m" desde la ventana de comandos del propio programa o se utiliza la aplicación ejecutable (Standalone Application).

presentacion.m

Este archivo ejecuta el sistema estableciendo unas condiciones iniciales de funcionamiento mediante la declaración y asignación de variables globales. Algunas de estas variables globales son las dimensiones de la máquina tuneladora, distancia entre representaciones gráficas, factor de escalado, etc. Además, una vez establecidas dichas condiciones de inicio, ejecuta el programa principal de la interfaz gráfica "panel_total.m".

panel_total.m

Esta función es la más importante ya que está continuamente ejecutándose debido a que está regida por un bucle infinito de repetición.

Es la encargada de controlar el flujo de operación de todo el sistema de monitorización. Está continuamente leyendo el archivo de intercambio de datos correspondiente, a la espera de variaciones de las variables entrantes de la diana láser "acceso" y "gamma" y del simulador de hinca de tubos "z_simulada" y "gamma_simulada", para permitir el acceso al resto de funciones del software. Su función es la de comportarse como un semáforo permitiendo el paso si se cumplen ciertas condiciones impuestas.

Para permitir el acceso de la primera condición impuesta que rige la ejecución del software debe producirse un incremento en el avance de la máquina tuneladora con respecto a su posición anterior ó producirse un giro superior a 1° sobre su eje longitudinal. Acto seguido, se obtendrá acceso a la monitorización de datos reales sólo si la diana ha activado previamente la variable "acceso=1" y se obtendrá acceso a la representación de datos simulados sólo si la se ha activado previamente la variable "simulador=1".

La condición que rige tanto la representación gráfica en dos dimensiones como la representación gráfica en tres dimensiones es la misma. Por tanto, una vez que la variable "acceso=1" o "simulador=1", se podrán monitorizar representaciones reales o simuladas respectivamente en dos y tres dimensiones.

Una vez que el operario establezca en la interfaz gráfica externa la monitorización de una representación en dos dimensiones o en tres dimensiones y se cumplan las condiciones anteriormente detalladas, se ejecutarán las condiciones gráficas y su función correspondiente.

A continuación, se muestra en la figura 57 el diagrama de flujo de la función "presentacion.m" y "panel_total.m" anteriormente detalladas.

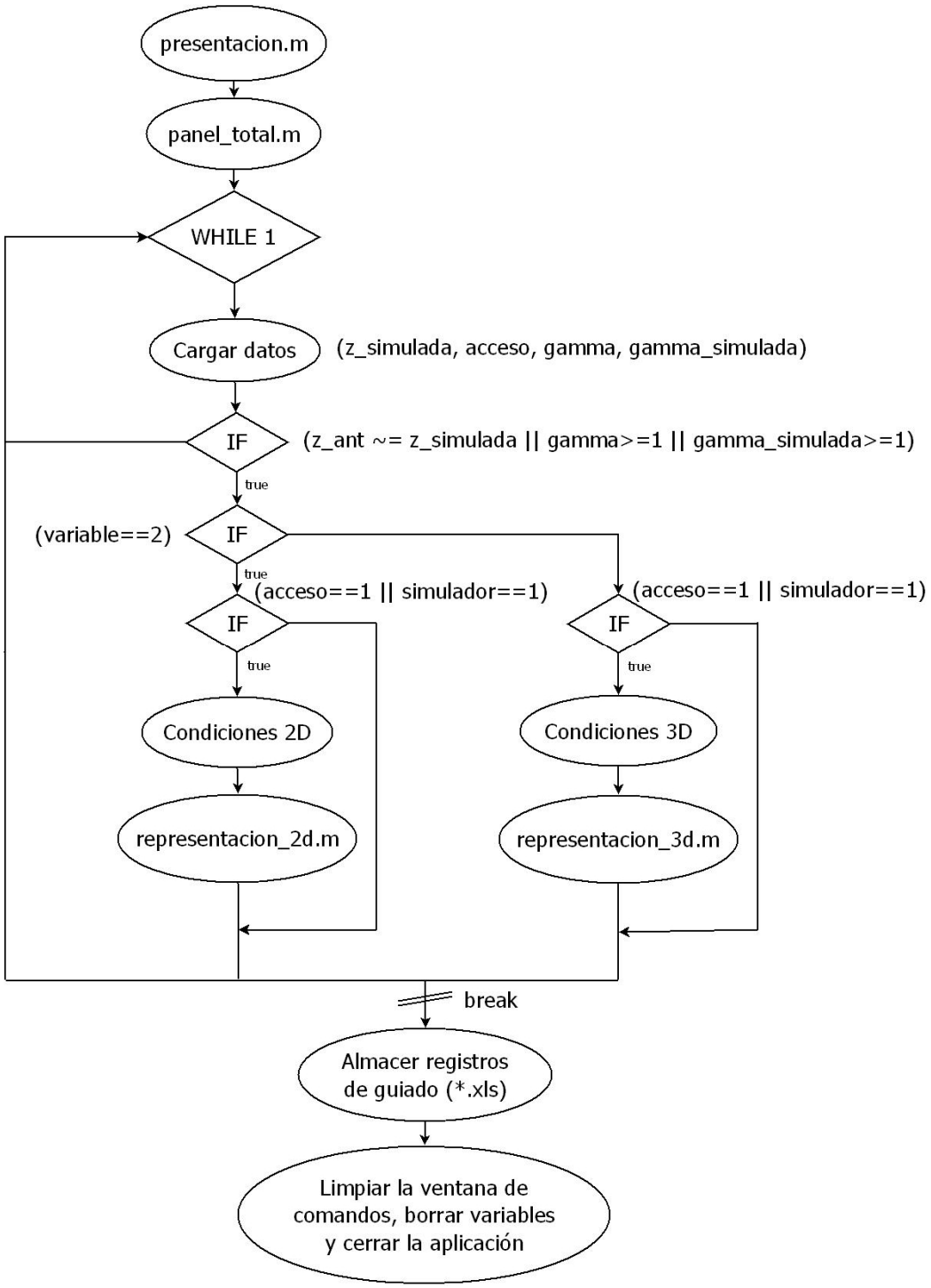


Figura 57. Diagrama de flujo de la función "presentacion.m" y "panel_total.m"

A continuación, se va explicar la función de representación de dos dimensiones "representacion_2d.m" y la función de representación de tres dimensiones "representacion_3d.m".

representacion_2d.m

Esta función se encarga en primer lugar, mediante la variable "acceso", de permitir el acceso a los datos reales procedentes de la diana o a los datos simulados procedentes del simulador de hinca de tubos.

- **Datos reales**

Si los datos que debe utilizar son los de la diana "acceso=1", cargará todas sus variables reales del archivo de intercambio de datos y calculará los parámetros necesarios para la representación gráfica de la flecha simbólica que tipifica la máquina tuneladora. Acto seguido, algunos parámetros calculados son guardados en el mismo archivo de intercambio de datos para más tarde poder enviarlos al simulador de hinca de tubos.

Si es el primer avance que se ha producido en la excavación, almacenará todos los parámetros calculados en las celdas de datos "A y C" y almacenará en la celda de datos "R" únicamente aquellos parámetros que serán expuestos en la tabla de registro de guiado de la interfaz gráfica externa.

Por el contrario, si ya se ha realizado el primer avance de la excavación, estos parámetros también quedan almacenados en las mismas tres celdas de datos "A, C y R" pero con algunas peculiaridades.

La celda de datos "A" es una matriz donde se van almacenando todos los parámetros y se va incrementando el número de filas según van produciéndose avances de la máquina tuneladora.

Sin embargo, la celda "C" tiene disponible cinco filas para guardar los últimos cinco avances que se produzcan en la excavación, con la condición que deben estar distanciados entre ellos al menos una distancia por sus respectivas variables de avance "z". Esta distancia es un tercio de las dimensiones totales de la máquina tuneladora y viene marcada por la variable "limite", que su valor depende del factor de escalado elegido en la interfaz gráfica.

La celda "R" almacena únicamente los parámetros correspondientes para la exposición de la tabla del registro de guiado de la misma manera que se van almacenando los parámetros en la celda "A".

Una vez almacenadas todas los parámetros reales en sus respectivas celdas, se vuelve a guardar en el archivo de intercambio de datos la variable "acceso=0" para que se cierre el acceso a la representación gráfica real hasta el próximo avance de la máquina tuneladora.

- **Datos simulados**

Si los datos que debe utilizar son los del simulador de hinca de tubos "simulador=1", cargará todas sus variables simuladas del archivo de intercambio de datos y calculará los parámetros necesarios para la representación gráfica de la flecha simbólica que tipifica la máquina tuneladora. Estos nuevos parámetros calculados no serán guardados en el archivo de intercambio de datos, ya que no será necesario enviarlos al simulador de hinca de tubos.

Si es el primer avance que se ha producido en la excavación, almacenará todos los parámetros calculados en las celdas de datos "B y D" y almacenará en la celda de datos "S" únicamente aquellos parámetros que serán expuestos en la tabla de registro de guiado de la interfaz gráfica externa.

Del mismo modo, si ya se ha realizado el primer avance de la excavación, estos parámetros calculados también quedan almacenados las tres celdas de datos.

En la celda de datos "B" se van almacenando los parámetros incrementalmente según se van produciéndose avances en la excavación.

La celda "D" tiene disponible cinco filas para guardar los últimos cinco avances que se produzcan en la excavación, con la condición que deben estar distanciados entre ellos al menos una distancia por sus respectivas variables de avance "z_simulada", del mismo modo regidas por la variable "limite" anteriormente comentada.

Por último, la celda "S" donde almacena únicamente los parámetros correspondientes para la exposición de la tabla del registro de guiado.

Posteriormente, una vez que las celdas han almacenado correctamente todos los parámetros calculados, la función dependiendo de si está trabajando con datos reales o simulados, carga los parámetros calculados de las matrices "A" o "B" y adecúa las condiciones de la interfaz gráfica externa para la apropiada monitorización gráfica de la flecha simbólica.

Por último, la función lleva a cabo la representación gráfica en dos dimensiones de la flecha simbólica y muestra por la pantalla de la interfaz gráfica los parámetros necesarios para el correcto guiado de la máquina tuneladora.

Es importante destacar que, tanto los datos reales procedentes de la diana como los datos simulados procedentes del simulador trabajan con la misma variable de avance "z_simulada". Esto es así, debido a que la diana láser no puede obtener la variable de avance real ya que esta distancia se calcula mediante una rueda métrica instalada en la parte superior delantera de la máquina tuneladora.

En la figura 58, se muestra el diagrama de flujo de la función “representacion_2d.m” encargada de cargar tanto los datos reales de diana como los datos simulados del simulador de hinca de tubos. Controla el flujo de operación para poder guardar y representar los parámetros calculados dependiendo del estado en que se encuentre activada la interfaz gráfica externa.

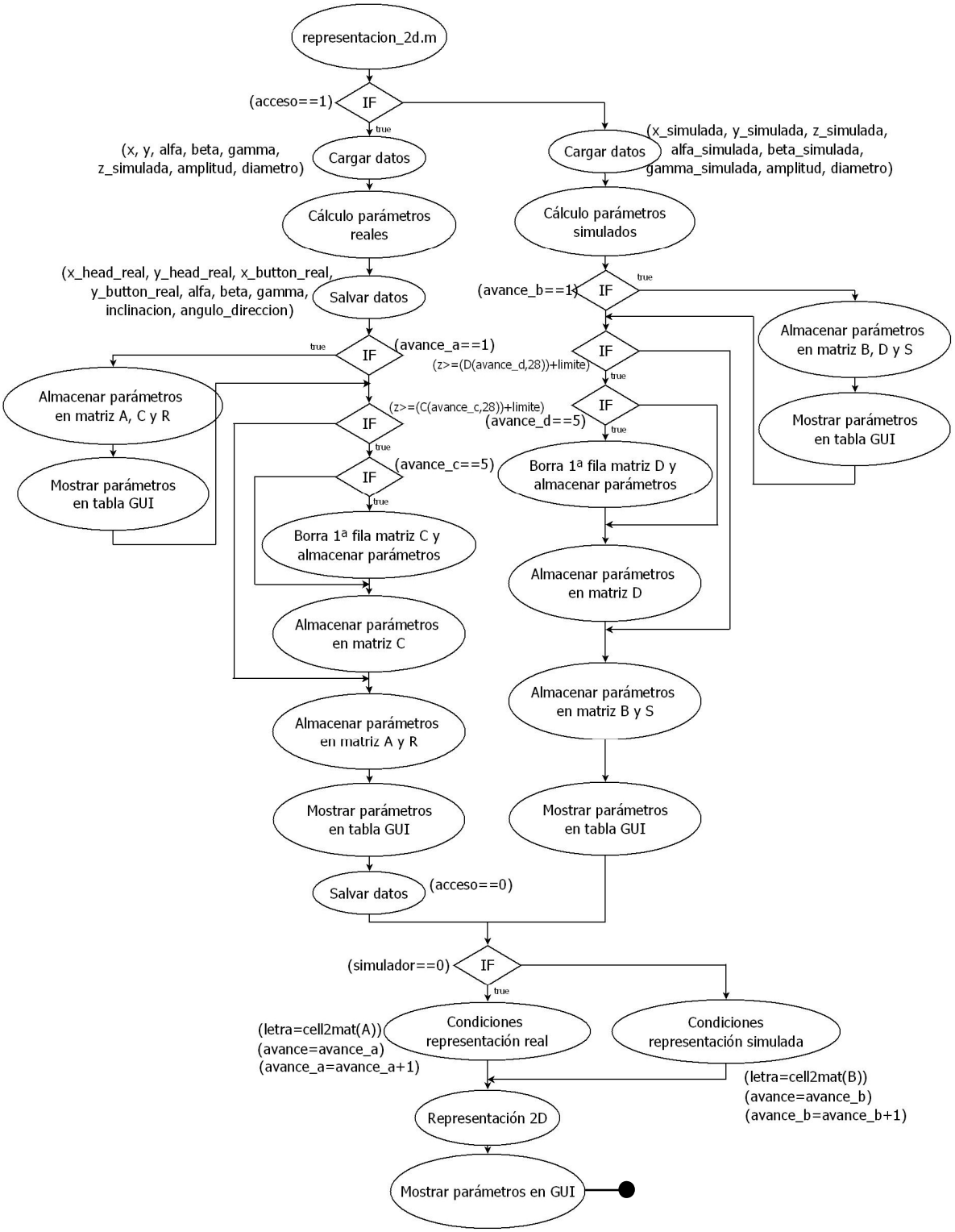


Figura 58. Diagrama de flujo de la función "representacion_2d.m"

representacion_3d.m

La estructura interna de la función es prácticamente igual que la función de representación en dos dimensiones explicada anteriormente pero con una única diferencia.

La función de representación en tres dimensiones, en primer lugar, realiza la distinción de datos reales o simulados mediante la variable "acceso".

En segundo lugar, calcula los parámetros necesarios para la representación gráfica y los almacena en sus correspondientes celdas "A", "B", "C", "D", "R" y "S" con el mismo criterio que la función de representación en dos dimensiones lo hacía anteriormente.

En tercer lugar, una vez que se han cargado las condiciones específicas para la monitorización gráfica de la flecha simbólica, la función entra en un bucle FOR que representa tantas flechas como filas halla almacenadas en la matriz "C" para los parámetros reales o en la matriz "D" para los parámetros simulados respectivamente.

Por tanto, si en alguna de las matrices "C" o "D" se encuentran almacenados cinco filas de parámetros con su correspondiente distanciamiento mínimo en sus variables de avance "z" o "z_simulada", se representarán gráficamente cinco flechas yuxtapuestas en la misma gráfica para poder apreciar los cambios de trayectoria y desviaciones angulares que se producen en el guidado de la máquina tuneladora.

A continuación se muestra un gráfico esquemático para la mejor comprensión de los parámetros que almacenan las celdas para cada avance que se produce en la excavación por medio de la máquina tuneladora. Véase figura 59.

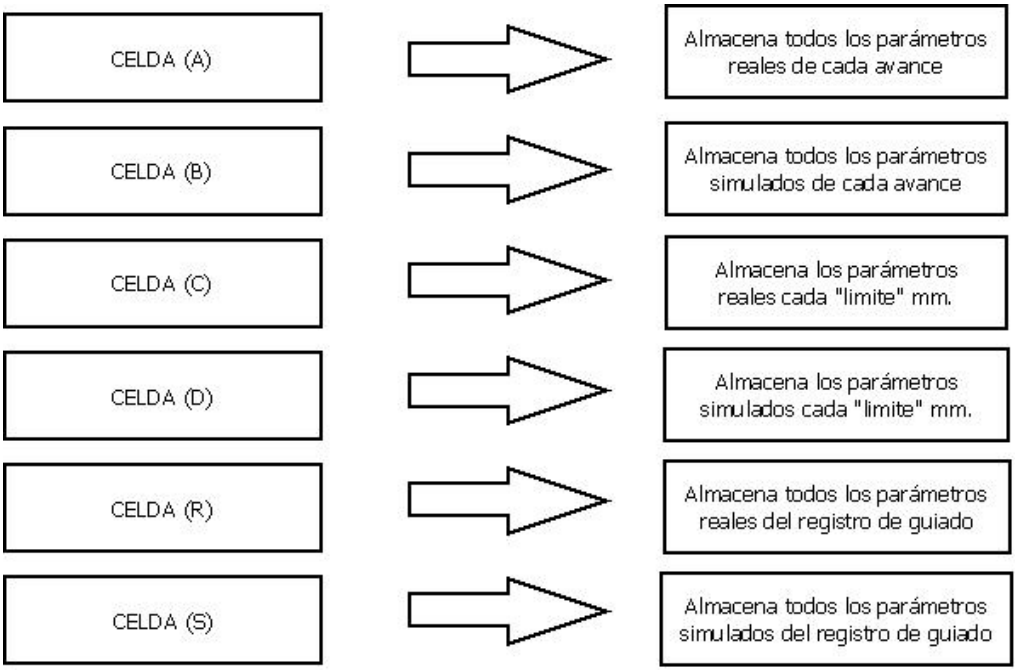


Figura 59. Nombre de celda donde se almacenan los diferentes tipos de parámetros

En la figura 60, se muestra el diagrama de flujo de la función "representacion_3d.m" encargada de cargar tanto los datos reales de la diana como los datos simulados del simulador de hinca de tubos. Controla el flujo de operación para poder guardar y representar los parámetros calculados dependiendo del estado en que se encuentre activada la interfaz gráfica externa.

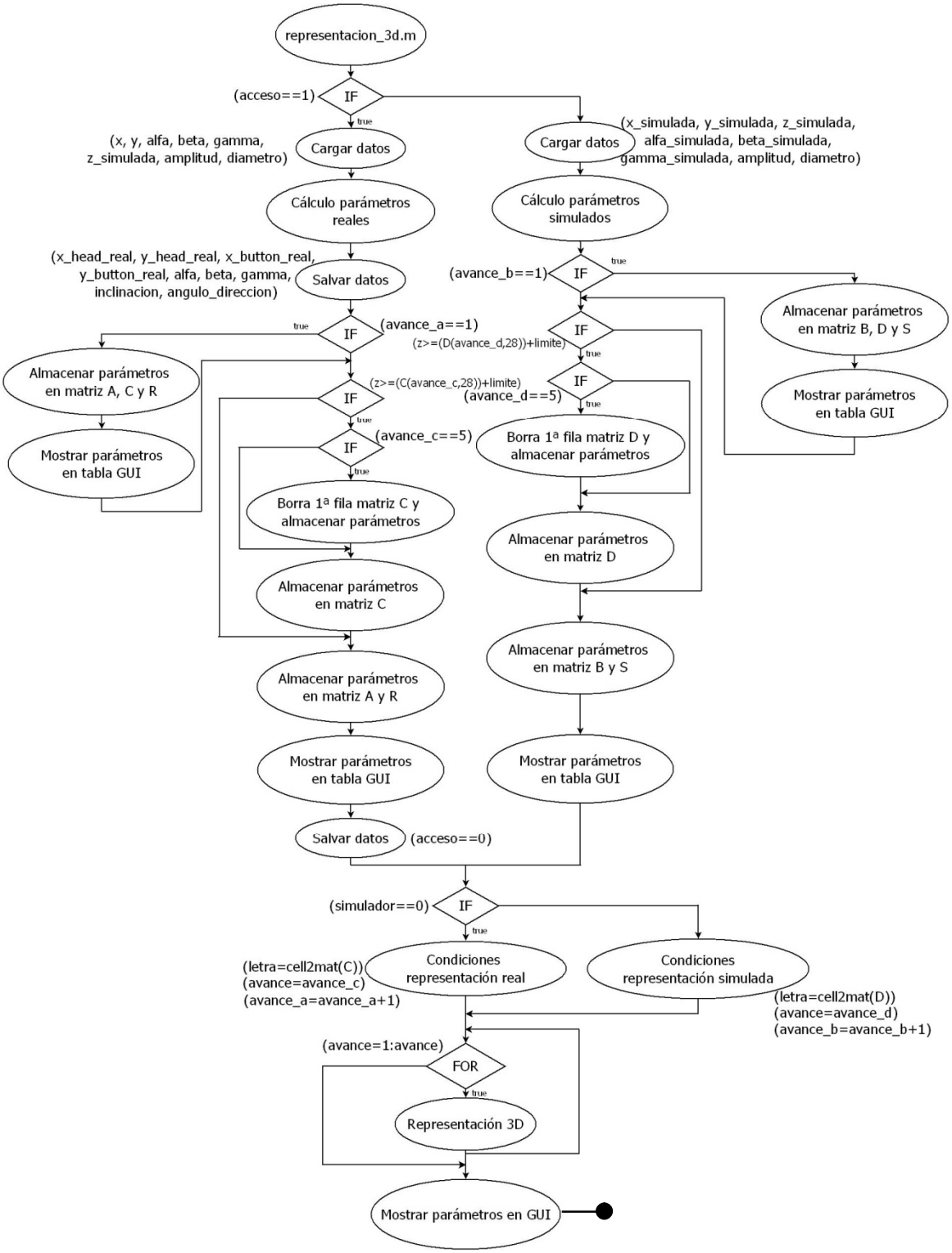


Figura 60. Diagrama de flujo de la función "representacion_3d.m"

En la figura 61, se muestra el diagrama de flujo que presenta todas las funciones de cálculo que realiza el software de la interfaz gráfica para hallar todos los parámetros necesarios para realizar la representación gráfica real o simulada de la flecha simbólica que tipifica la máquina tuneladora. A su vez, calcula todos los parámetros necesarios para monitorizar en la interfaz gráfica.

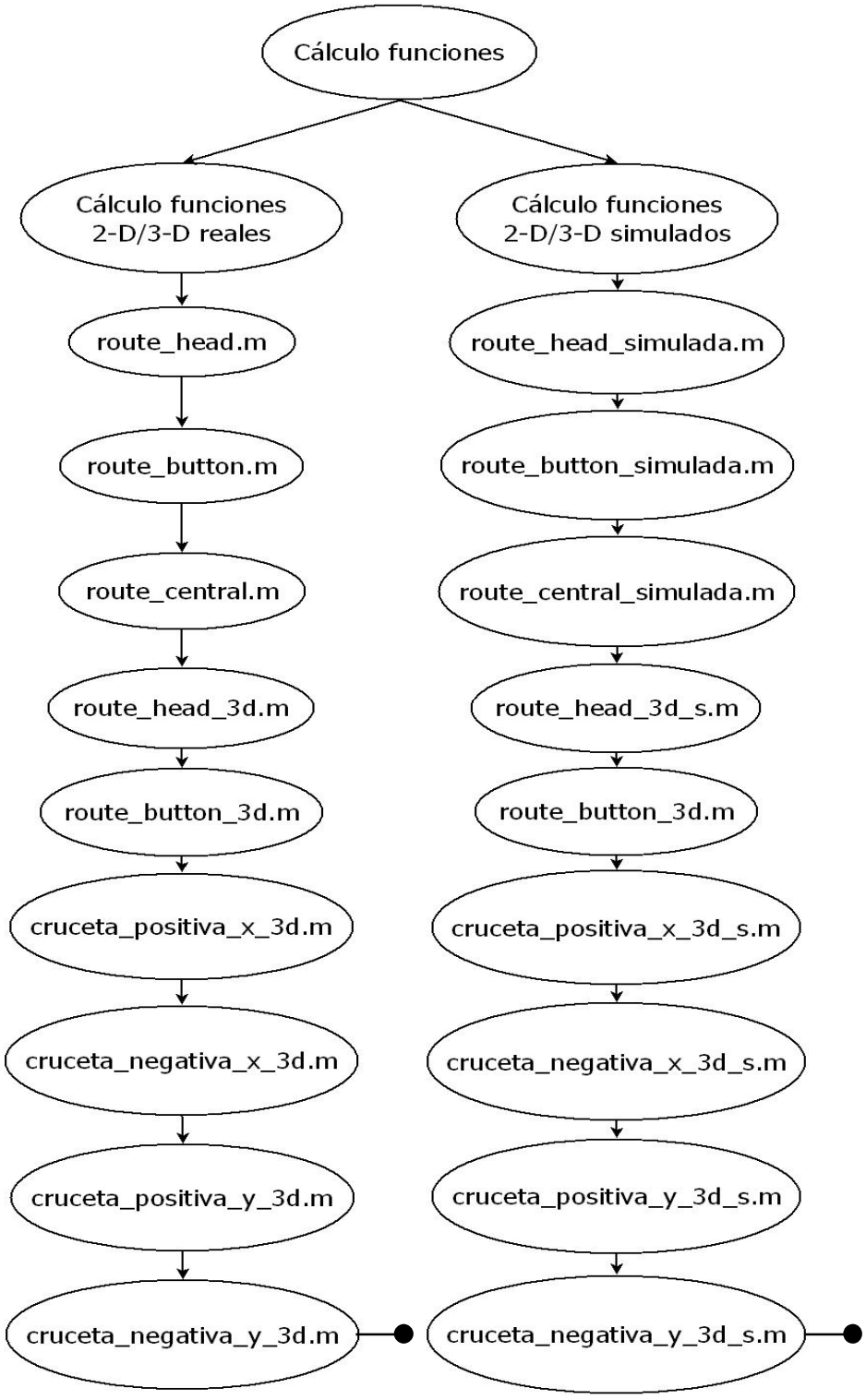


Figura 61. Diagrama de flujo de las funciones de cálculo de datos reales y simulados

3.4.2 Programación del software

La programación del software está estructurada por ficheros **.m* que están anidados entre sí y se van ejecutando unos u otros dependiendo de los datos recibidos a través del software de la diana, del simulador de hinca de tubos y del manejo del operario de la interfaz gráfica.

La descripción esquemática y el funcionamiento básico del sistema se han descrito anteriormente en el apartado 3.4.1.

A continuación, se explicará de forma detallada aspectos más internos de la programación del código para la compresión más sencilla de todas las funciones y comandos que puede llegar a realizar el software de la interfaz gráfica.

Declaración y asignación de variables globales

En primer lugar, cabe destacar la importancia de la declaración y asignación de variables globales a la hora de programar. Se declaran al principio de todos los ficheros **.m* donde se van a necesitar utilizarlas y su valor es visible en todos los espacios de trabajo de las funciones donde han sido declaradas.

Son muy útiles para realizar una programación rápida pero no se debe de abusar de ellas ya que disponen de un espacio de trabajo exclusivo y consumen muchos recursos. Esto provoca una pérdida de rendimiento provocando un funcionamiento ralentizado de la aplicación.

```
%--------------Declaración de variables globales--------------%
global pausa salir;
global A B C D R S;
global avance
global avance_a avance_b avance_c avance_d;
global variable;
global acceso simulador;
%_____________________________________________________________%

%--------------Asignación de variables globales---------------%
A=cell(1,28);   B=cell(1,28);   C=cell(5,29);
D=cell(5,29);   R=cell(1,17);   S=cell(1,17);
pausa=0;        salir=0;
avance=1;
avance_a=1;     avance_b=1;     avance_c=1;     avance_d=1;
variable=2;
acceso=0;       simulador=0;
%_____________________________________________________________%
```

La asignación de variables globales se realiza de igual forma que las variables locales. Se establecen unas condiciones iniciales para dichas variables y el software las irá modificando según vaya sucediendo la ejecución del código.

Llamadas a funciones y funciones

En segundo lugar, hay que destacar tanto la importancia de las llamadas a las funciones como las propias funciones.

- **Llamadas a funciones**

  Las llamadas a funciones o funciones definidas se utilizan de la misma forma que las funciones propias de MATLAB, es decir, basta con llamar a la función con los parámetros necesarios desde la ventana de comandos o desde otra función o "script".

  Por ejemplo, las funciones definidas en el *Current Directory* "`route_head_3d.m`" o "`route_button_3d.m`", basta con realizar la llamada a la función con sus argumentos de entrada y salida desde cualquier otra función para que se ejecute su código.

```
%Llamada a la función de cálculo de x,y,z de la cabeza de la flecha
[x_head,y_head,z_head]=route_head_3d(x,y,z,alfa,beta,gamma);

%Llamada a la función de cálculo de x,y,z del culo de la flecha
[x_button,y_button,z_button]=route_button_3d(x,y,z,alfa,beta,gamma);
```

Valores de retorno

Argumentos de entrada

- **Funciones**

  La primera línea de un fichero que define una función tiene la forma:

```
function [lista de valores de retorno] = nombre (lista de argumentos)
```

donde "`nombre`" es el nombre de la función. Entre corchetes y separados por comas van las variables de salida o valores de retorno, y entre paréntesis también separados por comas las variables de entrada o argumentos.

Si no hay valores de retorno se omiten los corchetes y el signo igual. Si sólo hay un valor de retorno no hace falta poner corchetes. Tampoco hace falta poner paréntesis si no hay argumentos.

```
function [x_head,y_head,z_head]= route_head_3d(x,y,z,alfa,beta,gamma)

dist_x=0;
dist_y=0;
```

Variables locales "*dist_x*" y "*dist_y*"

Las variables definidas dentro de una función son variables locales, es decir, las variables que se crean dentro de la función pertenecen al espacio de trabajo de la función y son inaccesibles desde otras partes del programa. Además no interfieren con variables del mismo nombre definidas en otras funciones o partes del programa y desaparecen una vez finaliza la llamada a la función. Para que la función tenga acceso a variables que no han sido pasadas como argumentos es necesario declarar dichas variables como variables globales, tanto en el programa principal como en las distintas funciones que deben acceder a su valor.

Por lo tanto, una función tampoco puede acceder a las variables definidas en el espacio de trabajo o "Workspace". Para ejecutar una función es necesario que el directorio de trabajo coincida con el directorio donde se encuentra la función.

Una diferencia importante con la programación en C es que en MATLAB los argumentos de una función no se modifican nunca y los resultados se obtienen siempre a través de los valores de retorno, que pueden ser múltiples y matriciales. Tanto el número de argumentos como el de valores de retorno no son fijos, dependiendo de cómo se llama a la función. No hace falta calcular siempre todos los posibles valores de retorno de la función, sino sólo los que se espera obtener.

Esto tiene importantes consecuencias en términos de eficiencia y ahorro de tiempo de cálculo.

Ficheros de intercambio de datos

Este fichero tiene gran importancia ya que es el método de comunicación utilizado para relacionar los diferentes módulos software que forman el sistema de guiado.

Los comandos "save" y "load" utilizados en las funciones o "scripts" permiten guardar y cargar variables, matrices y vectores de forma selectiva en ficheros con un nombre especificado por el usuario.

Primeramente, se almacenan en el fichero **.mat* las variables, vectores o matrices utilizando el comando "`save`" y asignando un nombre a dicho fichero binario.

Posteriormente, utilizando el comando "`load`", se carga del fichero binario **.mat* todos los valores de las variables, vectores y matrices y pueden ser utilizados para realizar otros cálculos u operaciones.

Este fichero binario tiene extensión **.mat* y se crea utilizando el comando:

```
save filename var1 var2 var3 -append
```

donde "`filename`" es el nombre del archivo de intercambio de variables y "`var1`, `var2`, `var3`…" son los diferentes nombres de las variables que se desean guardar en dicho fichero.

El comando "-append" permite añadir nuevas variables al archivo de datos donde ya se encontraban las anteriores variables almacenadas. En nuestro caso, los archivo de intercambio de variables que se utilizan para la comunicación con el software de la diana y del simulador de hinca de tubos se llaman "entradas.mat", "salidas.mat", "diana.mat" y "gui.mat".

```
%------Guardar variables en el fichero de intercambio de datos------%

save salida.mat x_head_real y_head_real x_button_real y_button_real...
               alfa beta gamma inclinacion angulo_direccion -append

%_____________________________________________________________________%
```

Para recuperar las variables almacenadas en el archivo de intercambio de datos desde cualquier punto del programa se utiliza el comando:

```
load filename var1 var2 var3
```

donde "**filename**" es el nombre del archivo de intercambio de variables y "**var1**, **var2**, **var3**…" es el nombre de las diferentes variables que se desean cargar del fichero.

```
%-------Cargar variables del fichero de intercambio de datos--------%
load entrada.mat z_simulada amplitud diametro
load gui.mat x y alfa beta gamma
%_____________________________________________________________________%
```

Por tanto, para la comunicación de variables, vectores o matrices entre dos ficheros **.m* ya sean funciones o "scripts", se realiza mediante la creación de un fichero común con extensión **.mat*.

Proceso de almacenaje de las celdas "C" y "D"

En este apartado, se va detallar la programación que sigue el proceso de almacenaje de parámetros de la celda "C" para los datos reales y de la celda "D" para los datos simulados.

En primera instancia, si se encuentra en el caso tanto de datos reales como de datos simulados, una vez calculados sus respectivos parámetros por medio de sus funciones de cálculo y suponiendo que se ha superado el primer avance de la excavación, se procede al almacenaje.

A continuación, se muestra el fragmento de código para el proceso de almacenaje de los parámetros reales en la celda "C", aunque el código para el proceso de almacenaje de los parámetros simulados es el mismo pero en la celda "D".

```
%-----------Condiciones de almacenaje de la celda "C" --------------%

if (z>=cell2mat(C(avance_c,28))+limite)

            if(avance_c==5)
                C(1,:)=[];
                C(5,1:29)=cell(1,29);
                avance_c=avance_c-1;
            end

            avance_c=avance_c+1;

C(avance_c,1:29)={x_head,y_head,z_head,x_button,y_button,z_button,...
                  x_cruceta_positiva_x,y_cruceta_positiva_x,...
                  z_cruceta_positiva_x,x_cruceta_negativa_x,...
                  y_cruceta_negativa_x,z_cruceta_negativa_x,...
                  x_cruceta_positiva_y,y_cruceta_positiva_y,...
                  z_cruceta_positiva_y,x_cruceta_negativa_y,...
                  y_cruceta_negativa_y,z_cruceta_negativa_y,...
                  x_head_real,y_head_real,z_head_real,...
                  x_button_real,y_button_real,z_button_real, ...
                  x_central,y_central,z_central,z,gamma};
end
%_____________________________________________________________________%
```

El comando "`if (z>=cell2mat(C(avance_c,28))+limite)`" impone la condición de que si el punto kilométrico actual (z) es mayor o igual que el punto kilométrico anteriormente almacenado en la celda "C" más la variable "limite", acepta dicha condición.

La instrucción "`cell2mat`" es necesaria para realizar la conversión de valor celda de la matriz a valor matemático para poder realizar la comparación con la variable "z".

El comando "`C(avance_c,28)`" permite obtener el valor del punto kilométrico central (z) que se encuentra almacenado en la fila "`avance_c`", columna 28 de la celda "C". La variable "`avance_c`" representa las filas de la celda "C" y tiene un rango de valores de 1 a 5 dependiendo del número de filas almacenadas.

Por lo tanto, si dicha variable "`avance_c`" obtiene el valor de 5, la celda "C" se convierte en una matriz de 5 filas por 29 columnas y está completamente almacenada de parámetros de avances de la excavación con un margen de separación variable respecto a sus puntos kilométricos (z).

Acto seguido, si se cumple la primera condición y la variable "`avance_c==5`", se acepta la segunda condición establecida y el primer comando que se encontra es "`C(1,:)=[];`". Este comando es el encargado de borrar la primera fila de parámetros de la celda "C" y trasladar las restantes cuatro filas de parámetros a la posición de 1 a 4.

El siguiente comando "`C(5,1:29)=cell(1,29);`" declara una nueva fila de celdas en la posición 5 de la celda "C" con el fin de reservar espacio para el siguiente avance de la excavación que cumpla las condiciones anteriores.

Por último, se encuentra el comando "`avance_c=avance_c-1;`" cuya función es reducir en una unidad el contador de "`avance_c`" que está igual a 5 para evitar que en el siguiente avance la máquina tuneladora vuelva a entrar en la condición anteriormente detallada.

Si no cumple la condición de que la variable "`avance_c==5`", se incrementa en una unidad el contador de "`avance_c`" con el comando "`avance_c=avance_c+1;`" y se almacenan los parámetros calculados en la fila correspondiente de la celda "C" con el comando "`C(avance_c,1:29)={x_head,y_head,z_head,x_button,…};`".

Monitorización de la tabla de registros de guiado

Para llevar a cabo el proceso de monitorización de la tabla de registros en la interfaz gráfica externa, hay que distinguir primero si los parámetros calculados son reales o simulados para recopilarlos en su correspondiente celda de almacenaje.

Acto seguido, se calcula previamente la fecha y hora para introducirlas como variables en la tabla de registro de guiado.

El comando "`t=fix(clock);`", cuyo deber es mediante la función "`clock`" almacenar en la variable "t" una matriz con el año, mes, día, hora, minuto y segundo. El prefijo "`fix`" establece un redondeo de todos los elementos de la matriz al entero lo más cercano a cero.

La variable "`hora`" almacena sólo la cuarta, quinta y sexta posición de la matriz "`t`" que corresponde a la hora, minuto y segundo. Además, se concatenan como separadores de la hora el símbolo de ":" entre cada posición.

El comando "`date`" permite guardar en la variable "`fecha`" el año, mes y día pero con formato dd-mmm-yyyy.

```
%-----------------Variables con la fecha y hora---------------------%
t=fix(clock);

hora=['   ',num2str(t(1,4)),':',num2str(t(1,5)),':',num2str(t(1,6))];
fecha=[' ',date];
%_____________________________________________________________________%
```

A continuación, se almacena en la celda "R" o "S" únicamente los parámetros correspondientes para la tabla del registro de guiado. El proceso de almacenaje de parámetros es el mismo tanto para los datos reales como para los simulados.

La monitorización de la tabla de registros se realiza mediante el comando "`set(handles.uitable1,'Data',R);`" o "`set(handles.uitable1,'Data',S);`" permitiendo colocar los parámetros almacenados en las celdas directamente en la interfaz gráfica externa para su posterior visualización.

```
%---Almacenaje de los parámetros reales de la tabla de registro---%

 R(avance_a,1:17)={y_head_real x_head_real y_button_real ...
                   x_button_real z_head_real z_button_real...
                   z_central inclinacion angulo_direccion ...
                   amplitud diametro rad2deg(alfa) rad2deg(beta)...
                   rad2deg(gamma) avance_a hora fecha};
%_____________________________________________________________________%

%--------Monitorización de la tabla en la interfaz gráfica---------%

set(handles.uitable1,'Data',R);
%_____________________________________________________________________%
```

Entorno previo de representación en dos y tres dimensiones

Una vez que se cumplan las condiciones impuestas en la función "panel_total.m" para llevar a cabo una representación en dos o tres dimensiones, el comando "`axes(handles.axes1)`" crea unos ejes con posición arbitraria y asigna su identificador a los ejes de la interfaz gráfica.

Esta instrucción es de vital importancia para que todos los comandos siguientes que tengan relación con estos ejes estén asignados al identificador propio de los ejes de la interfaz gráfica.

El comando "`cla`" se encarga de limpiar los ejes y borrar cualquier representación gráfica anterior.

El comando "`view(2)`" modifica el punto de vista de los ejes y los coloca en la vista 2-D predeterminada. Esta instrucción es muy importante ya sino se realizara los cambios de visión de dos a tres dimensiones o viceversa se quedaría con un punto de vista fijo y no se visualizarían ni las representaciones en dos dimensiones ni las de tres dimensiones.

El comando "`grid on`" dibuja una cuadrícula sobre los ejes y el comando "`hold on`" mantiene la representación actual y todas las propiedades de los ejes.

Los comandos "`xlabel, ylabel, zlabel`" permiten poner etiquetas a los ejes x, y, z de la interfaz gráfica y el comando "`title`" permite poner título a la gráfica de la interfaz. Los comandos "`xlim, ylim, zlim`" permiten poner límites a los ejes x, y, z de la interfaz gráfica.

A continuación, se muestran el fragmento de código que se utiliza para configurar las condiciones de representación en dos dimensiones.

```
%---------------Entorno previo representación 2D-----------------%

axes(handles.axes1)
cla
view(2)    %Vista en dos dimensiones
grid on
hold on

%-----------------Diámetro de los ejes escalados-----------------%
diametro_ejes=6000/FE;


%-----------------Límites de los ejes escalados-----------------%
xlim([-(diametro_ejes/2) (diametro_ejes/2)])
ylim([-(diametro_ejes/2) (diametro_ejes/2)])


%-------------Representación del título y las etiquetas----------%
  if(simulador==0)
    title('Gráfica 2-D Real','FontSize',20,'FontWeight','bold')
  else
    title('Gráfica 2-D Simulación','FontSize',20,'FontWeight','bold')
  end

xlabel('Eje Horizontal [mm]','FontSize',12,'FontWeight','bold')
ylabel('Eje Vertical [mm]','FontSize',11,'FontWeight','bold')


%-----------------Representación mirilla eje X--------------------%
mirilla_ejex_x=[-(diametro_ejes/2) (diametro_ejes/2)];
mirilla_ejex_y=[0 0];

plot(mirilla_ejex_x,mirilla_ejex_y,'k','lineWidth',2.5);


%-----------------Representación mirilla eje Y--------------------%
mirilla_ejey_x=[0 0];
mirilla_ejey_y=[-(diametro_ejes/2) (diametro_ejes/2)];

plot(mirilla_ejey_x,mirilla_ejey_y,'k','lineWidth',2.5);

%_____________________________________________________________%

%_____________________________________________________________%
```

Para la representación gráfica de la mirilla que actúa como diana, en primer lugar, se declaran las variables locales que indican las dimensiones de ésta.

Acto seguido, se utiliza el comando "`axis`" para declarar que los ejes de la gráfica van a tener las dimensiones programadas en las variables locales anteriores.

La representación del eje "X" de la mirilla se realiza creando una variable que contenga las dos coordenadas "X" y otra variable que contenga las dos coordenadas "Y", que en este caso serían 0.

Se utiliza el comando "`plot`" para realizar un vector entre las cuatro coordenadas del eje "X" especificadas con una anchura indicada por la propiedad "`'lineWidth'`".

Del mismo modo, se realiza la representación gráfica del eje "Y" en los ejes de la interfaz.

A continuación, se muestran el fragmento de código que se utiliza para configurar las condiciones de representación en tres dimensiones.

```
%-----------------Entorno previo representación 3D-----------------%
axes(handles.axes1)
cla
view(3)    %Vista en tres dimensiones
grid on
hold on
%-----------------Diámetro de los ejes escalados-----------------%
diametro_ejes=6000/FE;

%-----------------Límites de los ejes escalados-----------------%
xlim([-(diametro_ejes/2) (diametro_ejes/2)])
zlim([-(diametro_ejes/2) (diametro_ejes/2)])

  if(simulador==0)
    title('Gráfica 3-D Real','FontSize',20,'FontWeight','bold')
  else
    title('Gráfica 3-D Simulación','FontSize',20,'FontWeight','bold')
  end

xlabel('Eje Horizontal [mm]','FontSize',12,'FontWeight','bold')
ylabel('Eje de avance [m]','FontSize',11,'FontWeight','bold')
zlabel('Eje     Vertical     [mm]','FontSize',12,'FontWeight','bold')
%_____________________________________________________________%
```

El entorno previo de la representación en tres dimensiones es prácticamente idéntico al entorno de la representación en dos dimensiones, exceptuando que no se representa la mirilla como en el caso anterior.

Además, se utiliza el comando "`view(3)`" que modifica el punto de vista de los ejes y los coloca en la vista 3-D predeterminada. Esta instrucción es muy importante ya sino se realizara los cambios de visión de dos a tres dimensiones o viceversa se quedaría con un punto de vista fijo y no se visualizarían ni las representaciones en dos dimensiones ni las de tres dimensiones.

Funciones de representación gráfica

En este apartado, se va explicar detalladamente el criterio y las diferentes posibilidades que pueden adoptar los ficheros de "representacion_2d.m" y "representacion_3d.m" en función de los avances que se van sucediendo en el proceso de tunelación.

Las funciones de representación gráfica de dos dimensiones son diferentes que las de tres dimensiones. Sin embargo, las funciones correspondientes a cada tipo de representación gráfica, son iguales para datos reales y para datos simulados.

- **Representación gráfica en dos dimensiones**

A continuación, se muestra el diagrama de flujo de las diferentes posibilidades y condiciones que pueden adoptar las funciones de representación de dos dimensiones. Dependiendo de las condiciones que asuman las variables, el código ejecutará una de las cuatro funciones de representación gráfica. Véase figura 62.

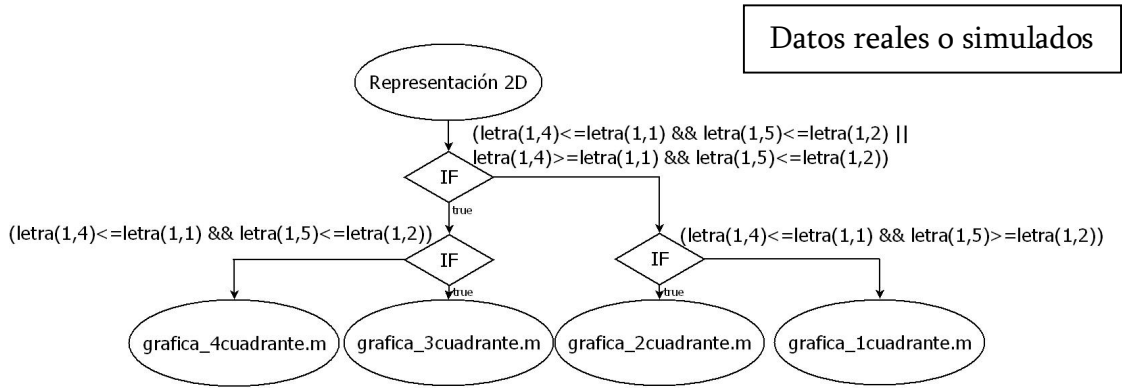


Figura 62. Diagrama de flujo de las condiciones de representación 2D

Antes de comenzar a explicar los criterios impuestos para la diversidad de funciones de representación gráfica y el trabajo que realiza cada una de éstas, es de especial importancia destacar la variable global "letra", que se utiliza en las condiciones impuestas para la selección de parámetros reales o simulados.

```
%----Almacenar en variable "letra" parámetros Reales o Simulados----%

if(simulador==0)
    letra=cell2mat(A);
    avance=avance_a;
else
    letra=cell2mat(B);
    avance=avance_b;
end
%_____________________________________________________________________%
```

En primera instancia, dependiendo de si se está trabajando con datos reales o simulados, la variable "`letra`" recibe todos los valores celda de la matriz "A" ó "B" y los transforma a valores matemáticos para poder trabajar con ellos. A continuación, la variable "`avance`" recibe el valor de la variable "`avance_a`" ó "`avance_b`" para saber en qué posición de avances reales o simulados se encuentra la excavación.

Acto seguido, es necesario conocer las variables que se almacenan en las columnas 1, 2, 4 y 5 de las celdas "A" y "B", con las que se está realizando las comparaciones y condiciones impuestas.

Las variables almacenadas en las columnas 1 y 2 de cada celda son los puntos (X, Y) de la cabeza de la máquina tuneladora respectivamente. Las variables almacenadas en las columnas 4 y 5 de cada celda son los puntos (X, Y) de la parte posterior de la máquina tuneladora respectivamente.

```
%Selección de función para representación 2D de la flecha simbólica-%

if(letra(avance,4)<=letra(avance,1) &&
   letra(avance,5)<=letra(avance,2) || ...
   letra(avance,4)>=letra(avance,1) &&
   letra(avance,5)<=letra(avance,2))

         if(letra(avance,4)<=letra(avance,1) &&
            letra(avance,5)<=letra(avance,2))

             grafica_3cuadrante;

         else

             grafica_4cuadrante;

         end

else

         if(letra(avance,4)<=letra(avance,1) &&
            letra(avance,5)>=letra(avance,2))

             grafica_2cuadrante;

         else

             grafica_1cuadrante;

         end

end
%_____________________________________________________________________%
```

Por lo tanto, las comparaciones que se producen tienen como objetivo elegir una representación gráfica **.m* u otra dependiendo de las posiciones delanteras y traseras que vaya adoptando la máquina tuneladora en el proceso de tunelación.

Las cuatro funciones que pueden ser elegidas dependiendo del cumplimiento de las condiciones establecidas son: "grafica_1cuadrante.m", "grafica_2cuadrante.m", "grafica_3cuadrante.m" y "grafica_4cuadrante.m".

El código interno de dichas funciones es exactamente igual entre todas. La única razón que las diferencia es que su código interno lleva el orden de ejecución alterado, es decir, dependiendo de la función en que se encuentre el programa se representarán unas partes de la flecha simbólica antes que otras.

La razón del diferente orden de representación entre unas funciones y otras es que dependiendo de los puntos, posiciones y ángulos de giro que vaya transmitiendo la máquina tuneladora, la visión de la flecha simbólica cambiará en función de la dirección que adopte.

A continuación, se muestran cuatro figuras ilustrativas que muestran las cuatro direcciones que puede adoptar la flecha simbólica y por consiguiente la correspondiente función **.m.* Véase figura 63, 64, 65 y 66.

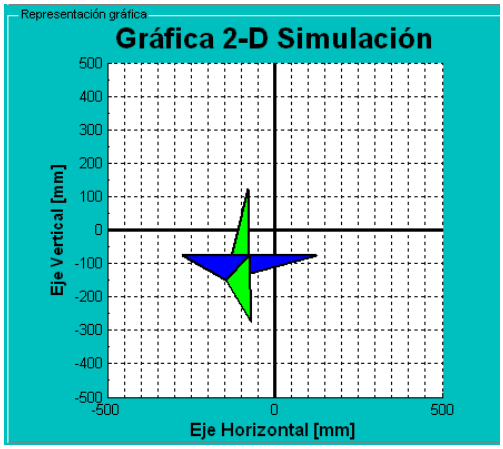


Figura 63. Dirección de la flecha para la función "grafica_1cuadrante.m"

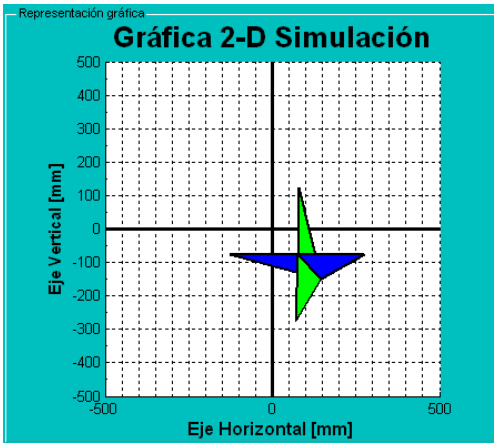


Figura 64. Dirección de la flecha para la función "grafica_2cuadrante.m"

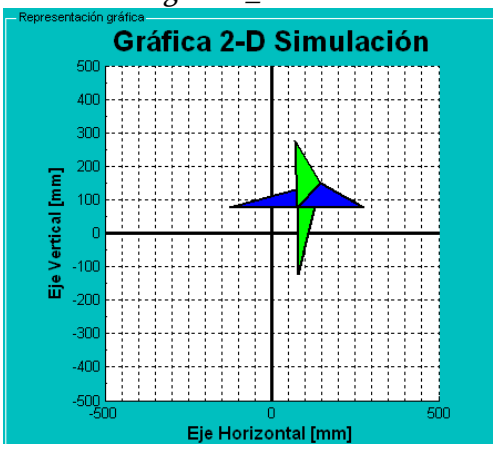


Figura 65. Dirección de la flecha para la función "grafica_3cuadrante.m"

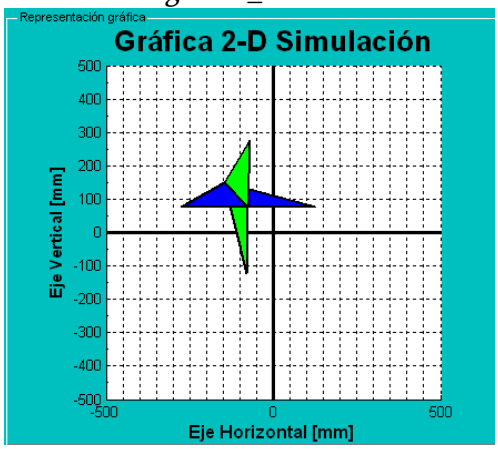


Figura 66. Dirección de la flecha para la función "grafica_4cuadrante.m"

Como se observa en las figuras anteriores, la flecha simbólica consta de cuatro triángulos rellenos de color verde o azul. Por tanto, dependiendo de la dirección que adopte ésta, se representarán gráficamente los triángulos en un orden determinado para que la flecha simbólica se visualice correctamente y pueda indicar claramente la dirección hacia la que apunta.

A continuación, se muestra un fragmento de código de la función "grafica_3cuadrante.m".

```
function grafica_3cuadrante

global letra;
global avance;

%Representación de la parte de abajo de la cruceta de la flecha
 poligono2_x=[letra(avance,4) letra(avance,1) letra(avance,16)];
 poligono2_y=[letra(avance,5) letra(avance,2) letra(avance,17)];
 fill(poligono2_x,poligono2_y,'g','lineWidth',2);

%Representación de la parte izquierda de la cruceta de la flecha
 poligono4_x=[letra(avance,4) letra(avance,1) letra(avance,10)];
 poligono4_y=[letra(avance,5) letra(avance,2) letra(avance,11)];
 fill(poligono4_x,poligono4_y,'b','lineWidth',2);

%Representación de la parte de arriba de la cruceta de la flecha
 poligono1_x=[letra(avance,4) letra(avance,1) letra(avance,13)];
 poligono1_y=[letra(avance,5) letra(avance,2) letra(avance,14)];
 fill(poligono1_x,poligono1_y,'g','lineWidth',2);

%Representación de la parte derecha de la cruceta de la flecha
 poligono3_x=[letra(avance,4) letra(avance,1) letra(avance,7)];
 poligono3_y=[letra(avance,5) letra(avance,2) letra(avance,8)];
 fill(poligono3_x,poligono3_y,'b','lineWidth',2);
```

Los comandos de la representación gráfica en dos dimensiones son:

- El comando "`fill`" rellena los polígonos de dos dimensiones del color especificado por la letra, en este caso, verde "`'g'`" o azul "`'b'`". Las variables "`poligonoX_x`" son matrices formadas por las tres coordenadas "X" del polígono a rellenar y las variables "`poligonoX_y`" son matrices formadas por las tres coordenadas "Y" del polígono a rellenar.

- El comando "`'lineWidth',2`" dibuja un vector negro que une los puntos dados por los polígonos (X,Y) y el número indica el grosor del vector. El color del vector que rodea a la flecha simbólica es el negro ya que se encuentra por defecto y no se indica otro color.

- **Representación gráfica en tres dimensiones**

  El criterio y las posibilidades que pueden adoptarse en la representación en tres dimensiones son exactamente las mismas que en la representación en dos dimensiones.

Acto seguido, es necesario conocer las variables que se almacenan en las columnas 1, 2, 4 y 5 de las celdas "C" y "D", con las que se está realizando las comparaciones y condiciones impuestas.

Las variables almacenadas en las columnas 1 y 2 de cada celda son los puntos (X, Y) de la cabeza de la máquina tuneladora respectivamente. Las variables almacenadas en las columnas 4 y 5 de cada celda son los puntos (X, Y) de la parte posterior de la máquina tuneladora respectivamente.

Es decir, la función "representacion_3d.m" mantiene todo el abanico de posibilidades y condiciones impuestas para elegir la función de representación gráfica más adecuada pero utiliza otras funciones de representación gráfica distintas a las anteriores.

Estas funciones son: "grafica_1cuadrante_3d", "grafica_2cuadrante_3d", "grafica_3cuadrante_3d" y "grafica_4cuadrante_3d". Véase figura 67.

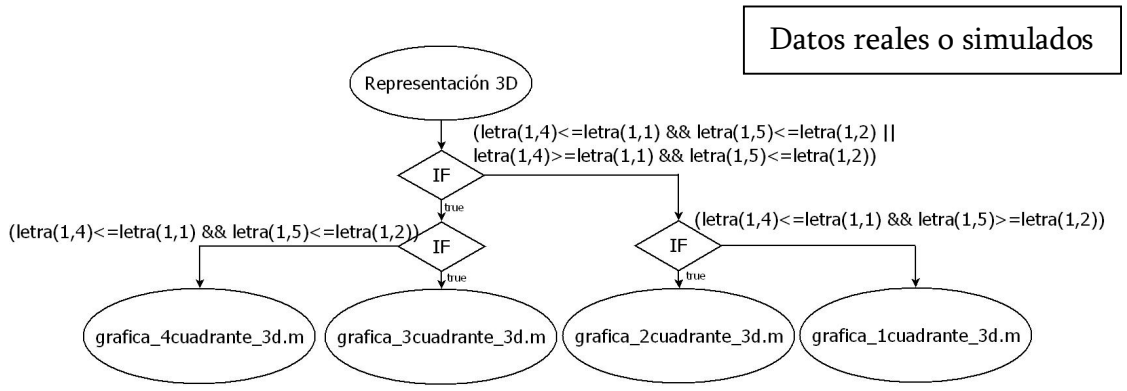


Figura 67. Diagrama de flujo de las condiciones de representación 3D

Estas funciones siguen el mismo orden de representación gráfica que las funciones de dos dimensiones. La diferencia entre ambas funciones es que la representación gráfica en tres dimensiones necesita trabajar con tres puntos o posiciones (X, Y, Z) delanteras y traseras de la máquina tuneladora para su representación gráfica mientras que la representación gráfica en dos dimensiones trabaja con dos posiciones (X, Y) delanteras y traseras.

A continuación, se muestra un fragmento de código de la función "grafica_2cuadrante_3d.m".

```
function grafica_2cuadrante_3d

global letra;
global avance;

%Representación de la parte de arriba de la cruceta de la flecha
 poligono1_x=[letra(avance,4) letra(avance,1) letra(avance,13)];
 poligono1_y=[letra(avance,5) letra(avance,2) letra(avance,14)];
 poligono1_z=[letra(avance,6) letra(avance,3) letra(avance,15)];
 fill3(poligono1_x,poligono1_z,poligono1_y,'g','lineWidth', 2);

%Representación de la parte izquierda de la cruceta de la flecha
 poligono4_x=[letra(avance,4) letra(avance,1) letra(avance,10)];
 poligono4_y=[letra(avance,5) letra(avance,2) letra(avance,11)];
 poligono4_z=[letra(avance,6) letra(avance,3) letra(avance,12)];
 fill3(poligono4_x,poligono4_z,poligono4_y,'b','lineWidth', 2);

%Representación de la parte de abajo de la cruceta de la flecha
 poligono2_x=[letra(avance,4) letra(avance,1) letra(avance,16)];
 poligono2_y=[letra(avance,5) letra(avance,2) letra(avance,17)];
 poligono2_z=[letra(avance,6) letra(avance,3) letra(avance,18)];
 fill3(poligono2_x,poligono2_z,poligono2_y,'g','lineWidth', 2);

%Representación de la parte derecha de la cruceta de la flecha
 poligono3_x=[letra(avance,4) letra(avance,1) letra(avance,7)];
 poligono3_y=[letra(avance,5) letra(avance,2) letra(avance,8)];
 poligono3_z=[letra(avance,6) letra(avance,3) letra(avance,9)];
 fill3(poligono3_x,poligono3_z,poligono3_y,'b','lineWidth', 2);
```

Otra peculiaridad es que los comandos utilizados para la representación gráfica en tres dimensiones son diferentes a los comandos utilizados en la representación gráfica de dos dimensiones.

Los comandos de la representación gráfica en tres dimensiones son:

- El comando "`fill3`" rellena los polígonos de tres dimensiones del color especificado por la letra, en este caso, verde "`'g'`" o azul "`'b'`". Las variables "`poligonoX_x`" son matrices formadas por las tres coordenadas "X" del polígono a rellenar, las variables "`poligonoX_y`" son matrices formadas por los tres coordenadas "Y" del polígono a rellenar y las variables "`poligonoX_z`" son matrices formadas por las tres coordenadas "Z" del polígono a rellenar.

- El comando "`'lineWidth',2`" dibuja un vector negro que une los puntos dados por los polígonos (X,Y,Z) y el número indica el grosor del vector. El color del vector que rodea a la flecha simbólica es el negro ya que se encuentra por defecto y no se indica otro color.

Monitorización paramétrica de datos reales y simulados

Para trasladar los parámetros calculados en las funciones a los campos de texto e indicadores visuales de la interfaz gráfica externa hay que hacer referencia al identificador de cada elemento en forma de "String". De esta manera, se puede referenciarlo al indicador correspondiente de la interfaz gráfica.

Tanto la función "representacion_2d.m" como la función "representacion_3d.m" utilizan los mismos identificadores ya que al compartir los dos modos de representación en la interfaz gráfica, deben hacer referencia a los mismos indicadores visuales y campos de texto.

```
%------Monitorización de los parámetros en la interfaz gráfica------%

set(handles.text1,'String',letra(avance,20));
set(handles.text2,'String',letra(avance,19));
set(handles.text5,'String',letra(avance,21));
set(handles.text3,'String',letra(avance,23));
set(handles.text4,'String',letra(avance,22));
set(handles.text6,'String',letra(avance,24));
set(handles.text7,'String',abs(letra(avance,20)-letra(avance,23)));
set(handles.text8,'String',abs(letra(avance,19)-letra(avance,22)));
set(handles.text9,'String',amplitud);
set(handles.text10,'String',diametro);
set(handles.text11,'String',avance);
set(handles.text12,'String',rad2deg(letra(avance,28)));

%------------------Monitorización de la hora y fecha----------------%

t=fix(clock);

set(handles.text13,'String',[num2str(t(1,4)),':',num2str(t(1,5)),':',
    num2str(t(1,6))]);
set(handles.text14,'String',date);
%_____________________________________________________________________%
```

Funciones de rotación y traslación

Hay que destacar una serie de funciones **.m* que tienen la misma estructura, funcionalidad y argumentos de entrada pero se diferencian en las variables locales utilizadas y por tanto, en sus argumentos de salida.

El objetivo principal de estas funciones es realizar traslaciones de puntos y rotaciones de ángulos para cambiar el sistema de referencia inicial impuesto por la diana de la máquina tuneladora hacia otros sistemas de referencia más útiles para poder trabajar con los parámetros con más facilidad.

A continuación, se muestra en la figura 68 un esquema gráfico de las dimensiones reales de la máquina tuneladora AVND 2000 AB y los diferentes sistemas de referencia hacia donde se van a trasladar y rotar los múltiples puntos y ángulos. Véase las dimensiones de la tunelarora AVND 2000 AB en Anexo A.

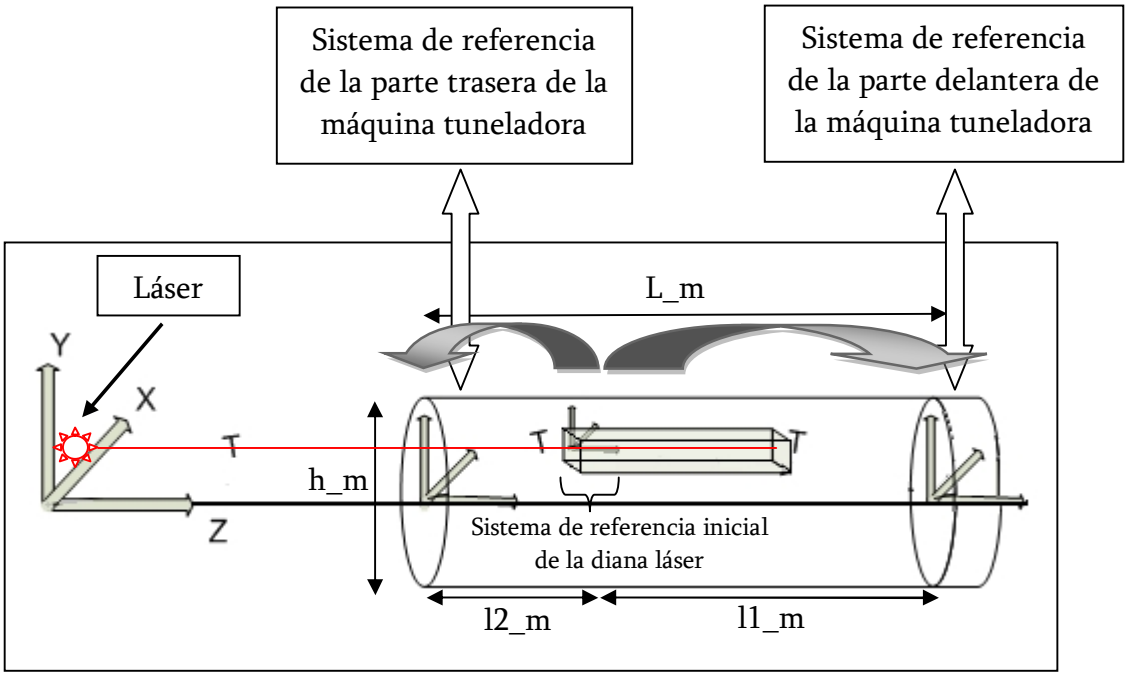


Figura 68. Dimensiones y sistemas de referencia de la máquina tuneladora y la diana

Las funciones que van a utilizar dicho esquema gráfico para realizar sus traslaciones de puntos y rotaciones de ángulos son: "route_head.m" y "route_head_simulada.m" hacia la parte delantera y "route_button.m" y "route_button_simulada.m" hacia la parte trasera de la máquina tuneladora.

El objetivo de los argumentos de salida de dichas funciones no va ser gráfico, sino meramente para la monitorización de los parámetros calculados en la tabla de registros de guiados y en los paneles de información paramétrica de la interfaz gráfica externa.

Sin embargo, se encuentran las funciones **.m* que calculan los parámetros de la flecha simbólica que tipifica la máquina tuneladora. Éstas también tienen que realizar traslaciones y rotaciones con respecto al mismo sistema de referencia global.

Por consiguiente, tanto las funciones que realizan cálculos con las dimensiones reales de la máquina tuneladora como las funciones que realizan los cálculos de la flecha simbólica hacen las traslaciones y rotaciones de coordenadas de la misma manera.

La diferencia se estriba en que las funciones que realizan los cálculos de la flecha simbólica trabajan con un factor de escalado introducido por el operario que reduce proporcionalmente las dimensiones originales de la máquina tuneladora.

Por tanto, sus funciones también tendrán que ser diferentes numéricamente a las funciones que calculan los parámetros de las dimensiones totales de la máquina tuneladora. Véase figura 69.

El objetivo de los argumentos de salida de dichas funciones escaladas va ser puramente gráfico ya que para la monitorización de los parámetros calculados en la tabla de registros de guiado y en los paneles de información paramétrica de la interfaz gráfica externa se va a utilizar las funciones que trabajan con las dimensiones reales de la máquina tuneladora.

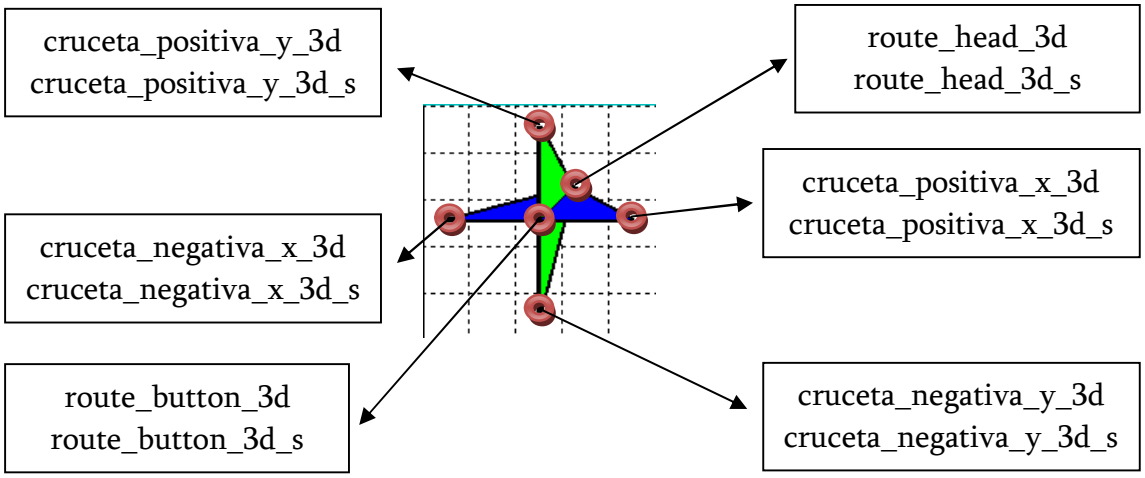


Figura 69. Funciones de cálculo reales y simuladas para cada punto de la flecha simbólica

En la figura 70, se muestra la flecha simbólica que tipifica la máquina tuneladora y las dimensiones escaladas de la máquina tuneladora real.

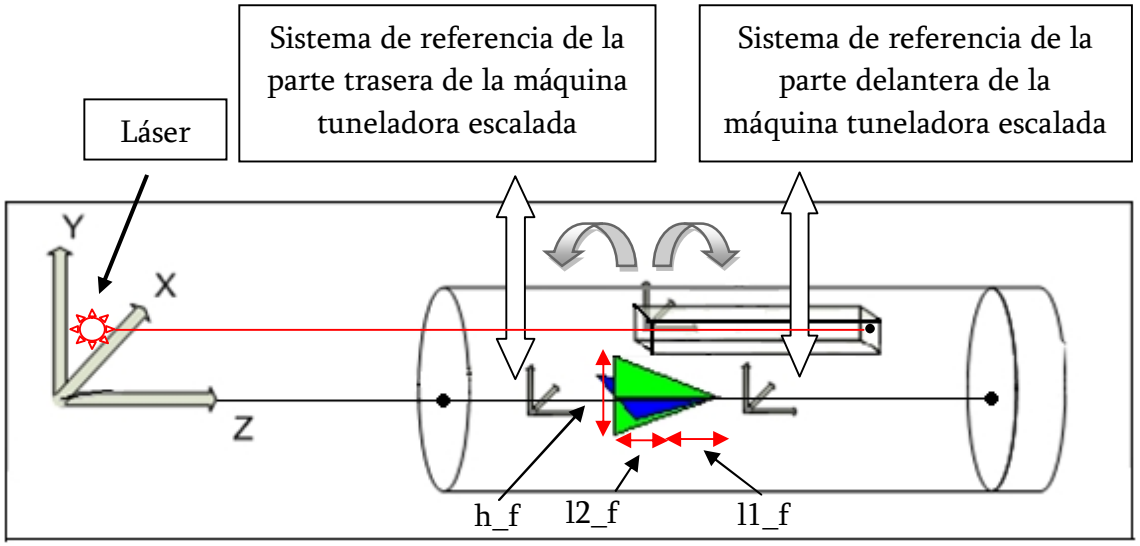


Figura 70. Dimensiones y sistemas de referencia de la flecha simbólica y la diana

A continuación, se va explicar detalladamente todo el proceso que hay que seguir para realizar los cálculos de coordenadas de la nueva posición girada y trasladada de la flecha simbólica a partir de los datos procedentes de la diana.

A partir de los datos recibidos por parte de la diana o del simulador de hinca de tubos, hay que calcular la nueva posición que ocupará la flecha simbólica después de tener en cuenta el desplazamiento horizontal y vertical (x, y) y sus ángulos de giro (alfa, beta y gamma).

La diana láser ha adoptado un punto de referencia situado en el panel incidente posterior de la diana para obtener las nuevas coordenadas con respecto al sistema de referencia global. Véase figura 71.

La programación que hay que seguir para realizar las transformadas de las coordenadas de la flecha simbólica, para datos reales procedentes de la diana láser o para los datos simulados procedentes del simulador, es la misma, con la diferencia que se utilizan otras funciones con sus respectivas variables.

Las funciones que intervienen en el cálculo de la nueva posición real de la flecha simbólica son las siguientes: "route_head_3d", "route_button_3d" "cruceta_positiva_x_3d", "cruceta_negativa_x_3d", "cruceta_positiva_y_3d", "cruceta_negativa_y_3d".

Para realizar las translaciones y rotaciones de las coordenadas de la flecha correctamente, bajo principios fundamentales de robótica, hay que seguir una secuencia de pasos en un orden determinado para conseguir los resultados esperados.

1. Realización de los desplazamientos y rotaciones angulares comenzando desde el sistema de referencia global situado en la entrada del túnel y referenciado a la altura del eje de simetría de la máquina tuneladora. Véase figura 70.

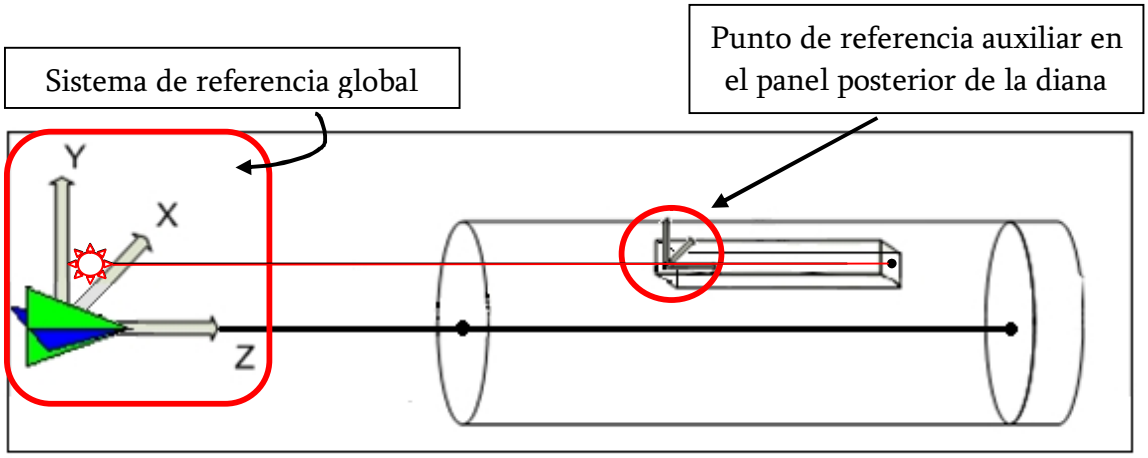


Figura 71. Punto de referencia auxiliar de la diana

2. Realización del giro del eje de avance "Z", para que se produzca sólo la rotación de las coordenadas de la flecha simbólica, ya que éstas no están desplazadas de su origen del sistema de referencia global. Véase figura 72.

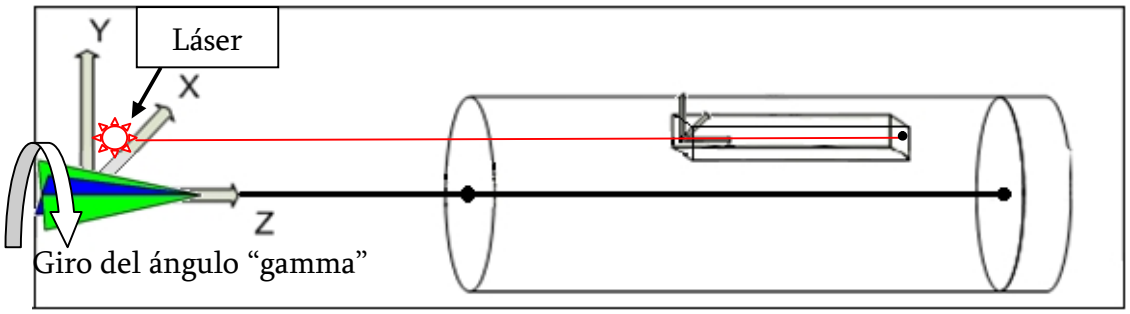


Figura 72. Rotación "gamma" de la flecha simbólica sobre su eje longitudinal

Para ello, se calcula la matriz de rotación "gamma" y se calcula la nueva posición de las coordenadas "pos", multiplicando dicha matriz de rotación por la matriz con las coordenadas de cada segmento que forma la cruceta de la flecha. En este caso, se ha elegido las coordenadas de la cruceta del eje "X" positivo. Por tanto, su función correspondiente es "cruceta_positiva_x_3d".

```
%--------------------------Giro gamma-----------------------------%
dist_x=202.08334;
dist_y=0;
dist_z=z-245.334;

G_z=rpy2tr(gamma*20,0,0);
pos=G_z*[dist_x;dist_y;dist_z;1];
```

3. A partir del avance dado "z" no proyectado, se calcula la proyección de la distancia que tendrá hasta la parte delantera o trasera de la flecha simbólica "z_proyeccion". Véase figura 73.

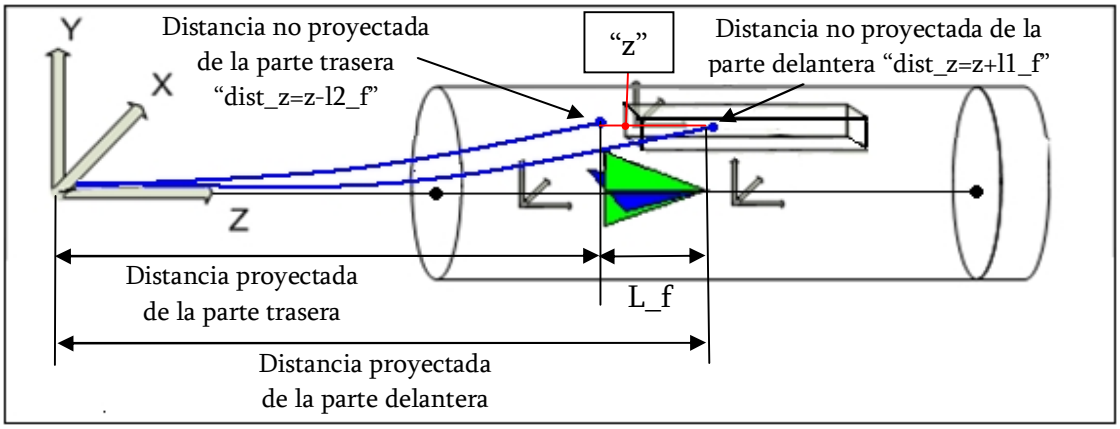


Figura 73. Distancia proyectada de la parte delantera o trasera de la flecha

Para ello, hay que realizar las rotaciones de "alfa" y "beta" y se calcula la nueva posición "z_proyeccion", multiplicando dicha matriz de rotación por la posición de las coordenadas de la parte delantera o trasera de la flecha. En este caso, se calcula la distancia a la parte posterior de la flecha debido a que las coordenadas de la cruceta se ubican en dicha parte.

```
%-------Cálculo de la proyección de z (delantera o trasera)----------%

dist_z=z-245.334;

G_XY=rpy2tr(0,beta,alfa);
z_proyeccion=G_XY*[0;0;dist_z;1];
```

4. Realización del giro del eje "X" y del eje "Y" para que se produzca la rotación de las coordenadas de la flecha simbólica. Antes de realizar el giro en "alfa" y "beta" de las coordenadas anteriormente giradas en "gamma" y se suman las coordenadas proyectadas provenientes de la diana "x" e "y". Véase figura 74.

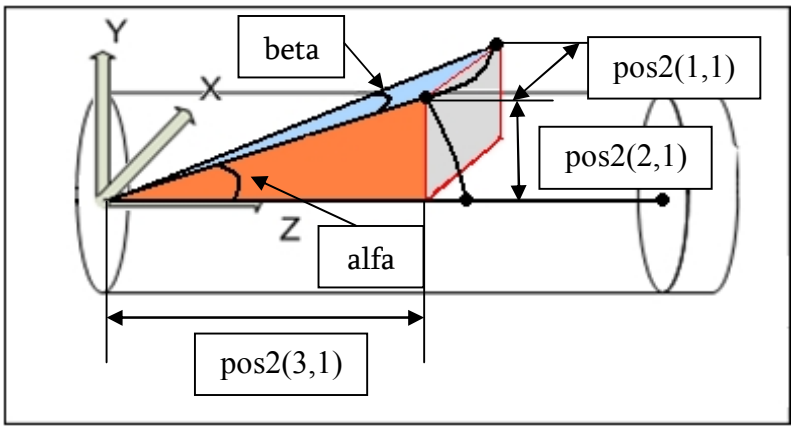


Figura 74. Rotación "alfa" y "beta" de la flecha sobre su eje vertical y trasversal

Para ello, se realiza la traslación de la distancia "z_avance" y se calcula la nueva posición de las coordenadas "pos2", multiplicando dicha matriz de traslación por la posición de las coordenadas actuales "pos".

```
%--------Rotación alfa y beta de las coordenadas proyectadas--------%

pos2=G_XY*[pos(1,1)+x;pos(2,1)+y;z_proyeccion(3,1);1];
```

Finalmente, se asocian las nuevas posiciones finales "x, y, z" a los valores de retorno de la función "cruceta_positiva_x_3d" y se divide la posición final "z" entre mil para convertirla de milímetros a metros.

Botones y menús de la interfaz gráfica

En las siguientes figuras se muestran los diagramas de flujo que recorre el software al interactuar el operario con la interfaz gráfica externa pulsando los botones disponibles y eligiendo los modos de representación gráfica a monitorizar.

En las figuras 75 y 76 se muestran los diagramas que actúan cuando el operario pulsa los botones de "detener" y "reanudar". El botón "Detener" cambia la variable "pausa" a 1. El software retorna a la línea de código justo anterior al momento de haber presionado dicho botón y continúa ejecutando el código con la modificación de la variable "pausa" establecida provocando la detección del mismo. Del mismo modo, el botón "Reanudar" cambia la variable "pausa" a 0 y retorna a la línea de código anterior a pulsar el botón, provocando la reanudación del software.

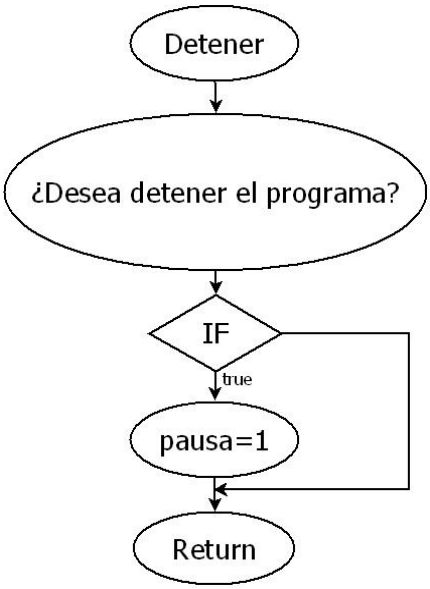


Figura 75. Diagrama del flujo del botón "Detener"

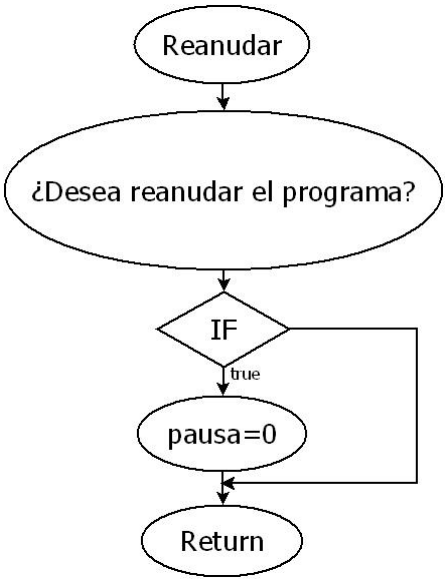


Figura 76. Diagrama de flujo del botón "Reanudar"

A continuación, se muestra el código de la funciones de atención a la interrupción de los botones de "Detener" y Reanudar" de la interfaz gráfica.

```
function detener_Callback(hObject, eventdata, handles)

global pausa
opc=questdlg('¿Desea detener el programa?','DETENER','Si','No','No');

   if strcmp(opc,'No')
      return;
   en
pausa=1;
```

```
function reanudar_Callback(hObject, eventdata, handles)

global pausa
opc=questdlg('¿Desea reanudar el programa?','REANUDAR','Si','No','No');

   if strcmp(opc,'No')
      return;
   end

pausa=0;
```

En la figura 77, se muestra el diagrama de bloques del botón “Salir”. Al pulsar dicho botón aparece un mensaje interrogativo que pregunta al operador si desea abandonar el programa. Si el operario cliquea la opción “No”, el software continúa ejecutándose normalmente sin realizar ningún cambio. Si por el contrario, el operario cliquea la opción “Si”, se produce un cambio en la variable “salir” a 1. El software retorna a la línea de código anterior y continúa ejecutándose con el cambio de variable establecido provocando la salida del programa.

Figura 77. Diagrama de flujo del botón "Salir"

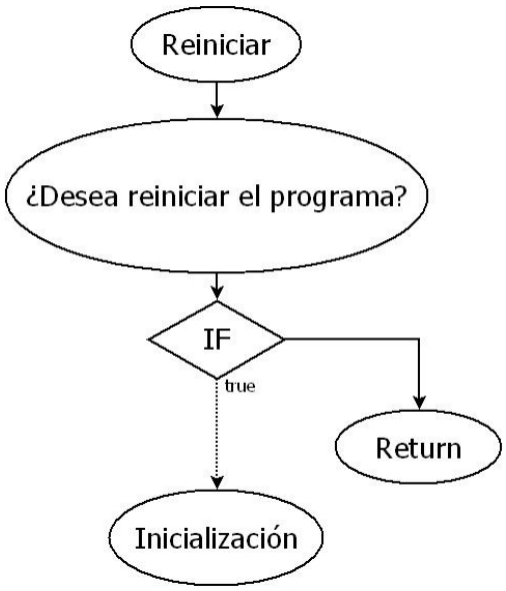


Figura 78. Diagrama de flujo del botón "Reiniciar"

En la figura 78, se muestra el diagrama de flujo del botón "Reiniciar". Al pulsar dicho botón aparece un mensaje interrogativo que pregunta al operador si desea reiniciar el programa. Si el operario cliquea la opción "No", el software continúa ejecutándose normalmente sin realizar ningún cambio. Si por el contrario, el operario cliquea la opción "Si", se produce la llamada a la función de inicialización "presentacion" reiniciándose de nuevo el software con las condiciones iniciales establecidas. A continuación, se muestra el código de la función de atención a la interrupción del botón de "Salir".

```
function salir_Callback(hObject, eventdata, handles)

global salir
opc=questdlg('¿Desea salir del programa?','SALIR','Si','No','No');

   if strcmp(opc,'No')
      return;
   end

salir=1;
```

A continuación, se muestra el diagrama de flujo de la acción que se ejecuta en el programa principal cuando se pulsan los botones de "Salir" o "Detener".

Cuando se pulsa el botón "Salir" (salir=1), se cumple la condición impuesta por el bucle "IF" y se ejecuta la sentencia "break". Esta instrucción hace que se rompa el bucle "WHILE 1" donde está englobada dicha sentencia de código y cerraría la aplicación. Véase figura 79.

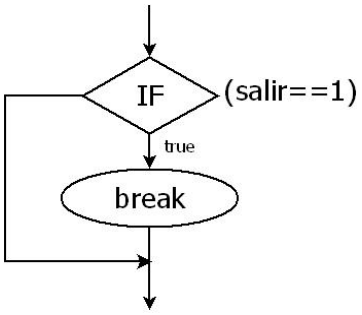


Figura 79. Diagrama de flujo de la condición de salida de la GUI

```
%---------------Condición de ruptura de bucle WHILE-----------------%
    if (salir==1)
        break
    end
%_____________________________________________________________________%
```

Cuando se pulsa el botón "Detener" (pausa=1) y no se pulsa el botón de "Salir" durante la pausa, se cumple la condición impuesta por el bucle "WHILE" y el programa entra en un estado de pausa continua hasta que cambien las condiciones impuestas. Si durante la pausa continua se pulsa el botón "Salir", rompería el bucle "WHILE" y entraría en el bucle "IF" anteriormente explicado. Véase figura 80.

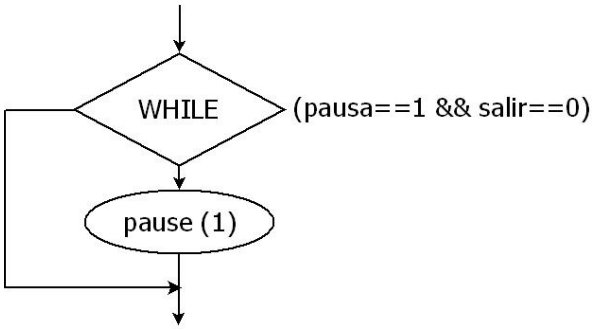


Figura 80. Condición de pausa y reanudación de la GUI

```
%-----------------Condición de pausa de la GUI----------------------%
    while (pausa==1 && salir==0)
        pause(1)
    end
%_____________________________________________________________________%
```

En las siguientes figuras, se muestra el diagrama de flujo de algunos de los menús contextuales que dispone la interfaz gráfica. De la misma manera que anteriormente hemos explicado el funcionamiento para los botones de "Detener" y "Reanudar", los siguientes botones "Representación Real / Gráfica 2-D", "Representación Real / Gráfica 3-D", "Representación Simulada / Gráfica 2-D" y "Representación Simulada / Gráfica 3-D" actúan con el mismo procedimiento.

Véase figuras 81 , 82, 83 y 84.

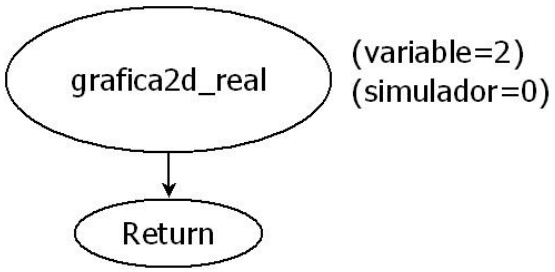


Figura 81. Diagrama de flujo del menú "grafica2d_real"

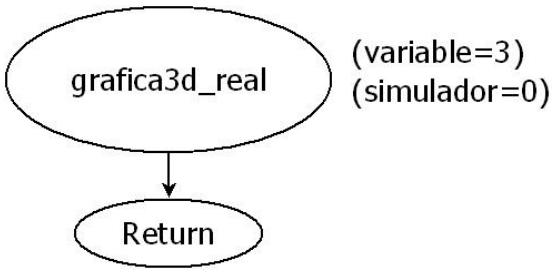


Figura 82. Diagrama de flujo del menú "grafica3d_real"

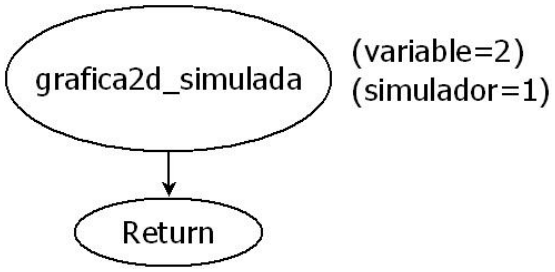


Figura 83. Diagrama de flujo del menú "grafica2d_simulada"

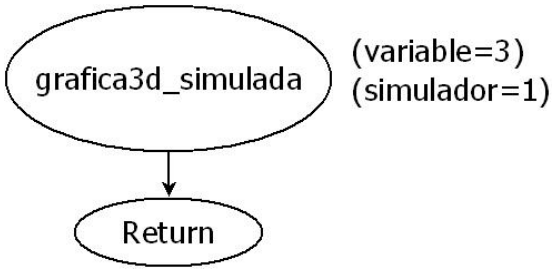


Figura 84. Diagrama de flujo del menú "grafica3d_simulada"

A continuación, se muestra el diagrama de flujo del componente de texto editable que se encuentra en la interfaz gráfica. Este texto editable lo utiliza el operario al introducir un factor de escalado. Este valor se almacena en una variable global para que pueda ser observado por todas las funciones de cálculo de parámetros y dimensiones del eje de la gráfica, y realizar los cambios correspondientes a dicho factor de escalado. Véase figura 85.

Figura 85. Diagrama de flujo del factor de escalado

El siguiente fragmente de código pertenece a la llamada de atención a la interrupción del componente texto editable. Se introduce en dicha función cuando el operario introduce algún valor por teclado en la interfaz gráfica para el factor de escalado. Lo almacena y lo transforma de 'String' a formato numérico 'double' y por último coloca el valor introducido en el mismo texto edible con formato 'String' de nuevo, para que refleje el valor actual en que se encuentra el factor de escalado.

```
%----------Se ejecuta cuando se introduce valor --------------------%
 function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global FE;
Val = get(handles.edit1,'String');
FE=str2double(Val);
set(handles.edit1,'String',FE);
%_____________________________________________________________________%
```

Por último, antes de abandonar el software pulsando el botón "Salir" de la interfaz gráfica externa, los parámetros reales y simulados monitorizados en las tablas de registro y almacenados en las matrices "R" o "S" son transcritos a un archivo Excel mediante el comando "`xlswrite`".

```
t=fix(clock);
fecha = date();

%-------Selección de la ruta de los registros reales y almacenaje------%
cd('C:\Documents and Settings\Usuario\Mis documentos\MATLAB\Labview\
Verano(11-08-2009)\Registros_reales')

xlswrite(strcat('Real-
',fecha,'_',num2str(t(1,4)),'·',num2str(t(1,5)),'·',num2str(t(1,6))),R);


%-----Selección de la ruta de los registros simulados y almacenaje-----%
cd('C:\Documents and Settings\Usuario\Mis
documentos\MATLAB\Labview\Verano(11-08-2009)\Registros_simulados')

xlswrite(strcat('Simulado-
',fecha,'_',num2str(t(1,4)),'·',num2str(t(1,5)),'·',num2str(t(1,6))),S);


%----------Selección de la ruta del directorio de trabajo--------------%
cd('C:\Documents and Settings\Usuario\Mis documentos\MATLAB\Labview\
Verano(11-08-2009)')
%_____________________________________________________________________%
```

Para finalizar el software, limpiar el espacio de trabajo de todas las funciones y cerrar la ventana de la interfaz gráfica externa se utiliza el comando "`clear,clc,close all;`".

```
%-------Limpiar variables, Command Window y cerrar aplicación--------%
clear,clc,close all;
```

3.5 Comunicación entre los diferentes módulos software

Para poder llevar a cabo la comunicación entre los diferentes módulos, es necesario el desarrollo tanto del hardware de todos los componentes pertenecientes al sistema de guiado como del software de los tres módulos relacionados, diana, simulador y GUI. Véase figura 86.

El conjunto de los tres módulos que forma el sistema de guiado completo, pretende tener un índice de repetitividad muy alto, en comparación con los sistemas de guiado actuales, para en primer lugar, poder contemplar en tiempo real la posición espacial donde se encuentra la máquina tuneladora y detectar con antelación sus posibles desviaciones y cambios de trayectoria.

En segundo lugar, tener un índice de repetitividad alto ayuda al operario a evitar posibles situaciones de peligro para la máquina tuneladora, como una obstrucción del cabezal cortante y el consiguiente giro sobre su eje longitudinal.

A parte de poder representar datos reales con la compenetración de los tres módulos anteriormente comentados, también se puede realizar simulaciones solamente con la comunicación del módulo del simulador y la GUI.

Esto es una gran innovación, ya que con las representaciones simuladas de la trayectoria ideal que se pretende conseguir, el operario puede familiarizarse al ambiente para intentar evitar con más facilidad los problemas que le puedan surgir cuando esté en una situación real.

Además, este simulador sirve como herramienta de aprendizaje para los futuros operarios que pretendar guiar la máquina tuneladora, sin tener al técnico especialista supervisando el manejo.

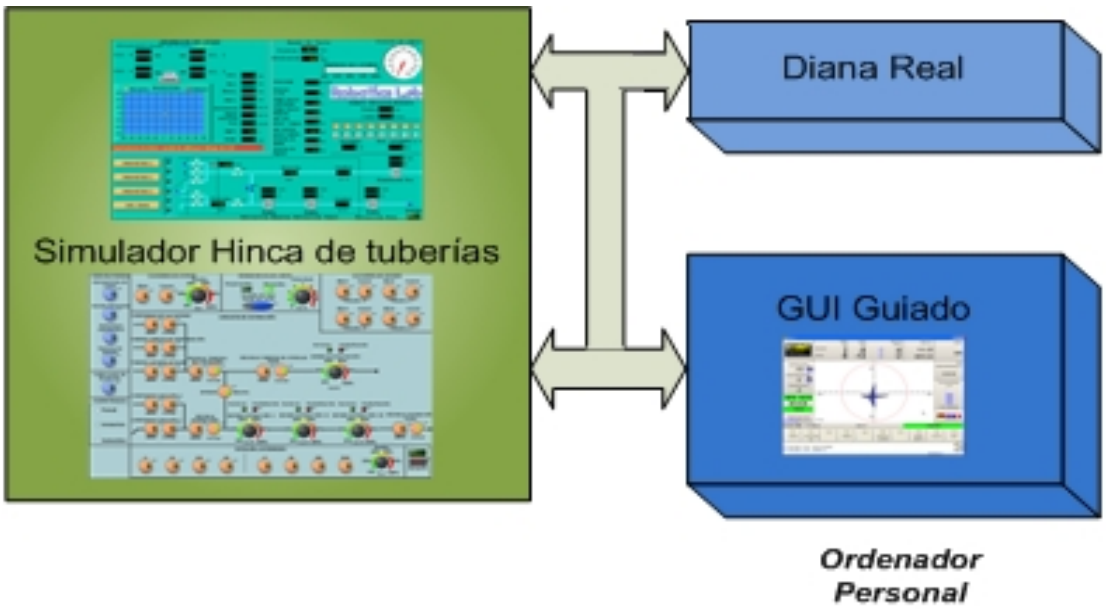


Figura 86. Diagrama de comunicaciones del sistema de guiado

Debido a ello, el sistema necesita de la comunicación entre sus partes, para poder hacer uso del mismo. Por ello, se ha desarrollado una comunicación entre los programas, donde se comparte la información que necesite cada aplicación en cualquier momento. En vista de la necesidad de cada aplicación de obtener valores de otros, se ha llegado a la necesidad de compartir las variables que se recogen en el siguiente gráfico. Véase figura 87.

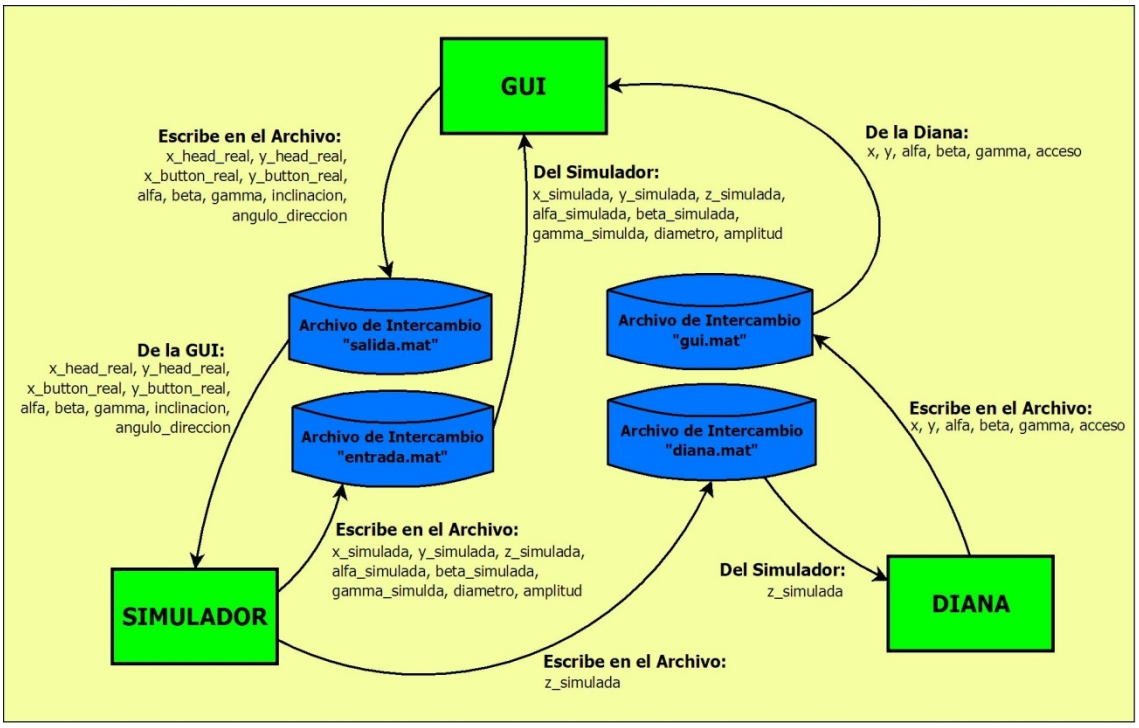


Figura 87. Comunicación entre los tres módulos y variables relacionadas mediante archivos de intercambio de variables

Tanto la diana como la GUI han sido desarrollados mediante el programa MATLAB, por lo que se han barajado distintas formas de comunicación comunes de este programa. El simulador está siendo desarrollado en el programa LabVIEW, lo que ha representado mayores problemas a la hora de obtener una forma rápida y fácil de comunicación.

Los métodos considerados para la comunicación entre los distintos módulos han sido:

- Variables globales entre programas.
- Realización de un archivo de intercambio de datos.
- Comunicación por Socket.
- Realización de una librería de enlace dinámico (DLL).

3.5.1 Alternativas de comunicación entre LabVIEW y MATLAB

3.5.1.1 Librería de enlace dinámico

Dynamic Link Library o más comúnmente DLL es el término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda de un programa por parte del sistema operativo.

Un archivo DLL contiene funciones que el programa ejecutable puede llamar durante la ejecución, es decir, un archivo DLL es una biblioteca de funciones que el programa puede vincular dinámicamente. Un vínculo puede ser estático o dinámico.

Toda la información de dirección necesaria por el programa para tener acceso a la función de biblioteca es fija cuando el archivo ejecutable se crea y permanece sin cambios durante la ejecución.

MATLAB permite usar las funciones de estas librerías que están programadas en C++ a través de una interfaz de línea de comandos. Esta interfaz ofrece la posibilidad de cargar una librería externa en MATLAB y acceder a cualquiera de las funciones definidas en dicha librería. Aunque los tipos de datos son diferentes en MATLAB y en C++, en muchos casos es posible pasar los tipos de datos de MATLAB a C++ sin tenerse que preocupar de la conversión porque MATLAB la realiza de forma automática.

Para crear un librería dinámica (**.dll)* y poder mantener comunicación e intercambio de datos con otras aplicaciones externas (LabVIEW), en primer lugar, es necesario crear una librería dinámica, que contenga todos los archivos de funciones "**.m"* de nuestra carpeta específica de trabajo de MATLAB utilizando para ello el compilador interno de MATLAB.

Acto seguido, se genera la función "wrapper" mediante el programa Visual Studio. La función "wrapper" es una función que crea un envoltorio alrededor de la librería dinámica anteriormente generada por MATLAB. El objetivo que tiene la creación de dicho envoltorio es la programación de una serie de declaraciones para poder acceder a las funciones de la librería dinámica compilada anteriormente desde MATLAB y a sus respectivas variables tanto de entrada como de salida, para que otras aplicaciones externas puedan intercambiar datos, comunicarse y acceder a las funciones de dicha librería dinámica.

El último paso es generar el archivo de librería dinámica definitivo, de extensión ".dll" utilizando el compilador propio de Visual Studio para compilar la función "wrapper". A continuación, se adjunta la librería dinámica generada por MATLAB a la nueva librería dinámica de la función "wrapper" para que ésta última pueda acceder a las funciones de la librería del programa y de esta manera mantener intercambio de variables con otras aplicaciones externas, en este caso, LabVIEW. Véase figura 88.

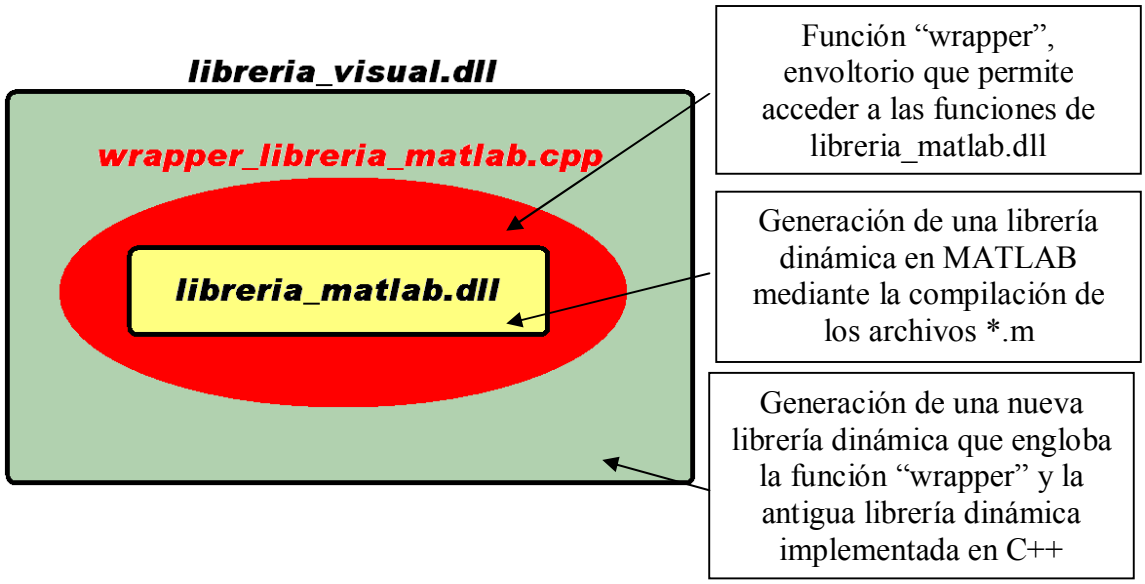


Figura 88. Gráfico de la composición de una librería dinámica

A continuación, se exponen detalladamente todas las instrucciones que hay que seguir para la generación de un archivo de librería dinámica para poder mantener comunicación e intercambio de datos con otros programas externos.

En primer lugar, hay que generar una librería dinámica (**.dll*) de los archivos **.m* de nuestro proyecto de MATLAB mediante el compilador propio de MATLAB ó el compilador de Visual Studio. Para ello, se selecciona la carpeta de trabajo en el *Current Directory*, se abre para explorar los archivos **.m* y a continuación se escribe en el *Command Window* el siguiente comando: *mbuild –setup*

Esta instrucción permite al usuario seleccionar entre el compilador de Visual Studio o el compilador de MATLAB para la compilación de la librería dinámica de nuestro software. Se puede escoger cualquier compilador de los dos, en este caso, se ha escogido el compilador de Visual Studio.

```
>> mbuild -setup
Please choose your compiler for building standalone MATLAB
applications:
Would you like mbuild to locate installed compilers [y]/n? y

Select a compiler:
[1] Lcc-win32 C 2.4.1 in C:\ARCHIV~1\MATLAB\R2008a\sys\lcc
[2] Microsoft Visual C++ 2008 in c:\Archivos de programa\Microsoft
Visual Studio 9.0
[0] None

Compiler: 2
```

A continuación, para generar la librería dinámica compartida de nuestros ficheros **.m* se ejecuta en el *Command Window* la siguiente sentencia:

*mcc -t -L C -W lib:libreria_matlab -T link:lib -h <archivos *.m> libmmfile.mlib*

La opción –t informa al compilador de MATLAB que traslade el código de los archivos **.m* a otro lenguaje. La opción –L especifica que el lenguaje elegido es el C. La opción –W informa al compilador de MATLAB que construya un envoltorio para las librerías con el nombre especificado después de "lib:". La opción –T informa al compilador que debe unir nuestra aplicación conjuntamente para construir una librería compartida.

Este comando creará un listado de archivos en el mismo directorio de trabajo del *Current directory* en el que se encontraba nuestros ficheros **.m*.

En segundo lugar, hay que crear la función "wrapper". Para ello, hay que seguir una serie de procedimientos con el programa Visual Studio.

1. Ejecutar Microsoft Visual Studio

2. File → New → Project…

3. Seleccionar Win 32 Console Application. Véase figura 89.

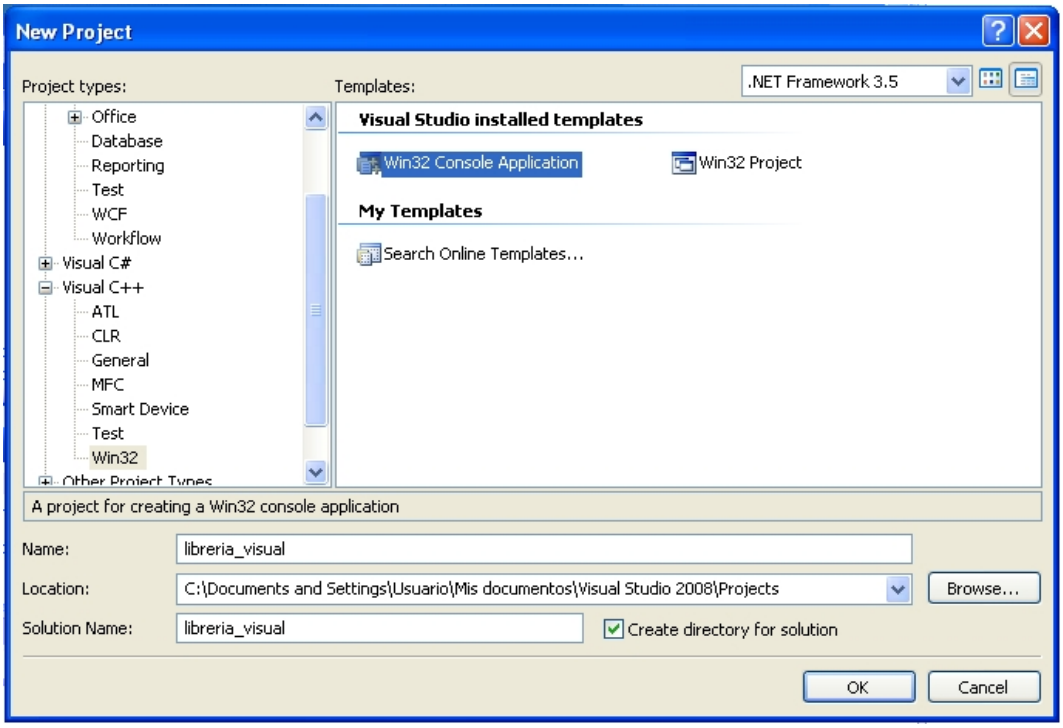


Figura 89. Selección del tipo de proyecto "Win32 Console Application"

4. Introducir un nombre para el proyecto, en este caso, "libreria_visual".

5. Presionar OK.

6. Elegir “Applications Settings” y dentro del menú “Application type” escoger la opción “DLL”. A continuación, dentro del menú “Additional options” escoger la opción “Export symbols”. Véase figura 90.

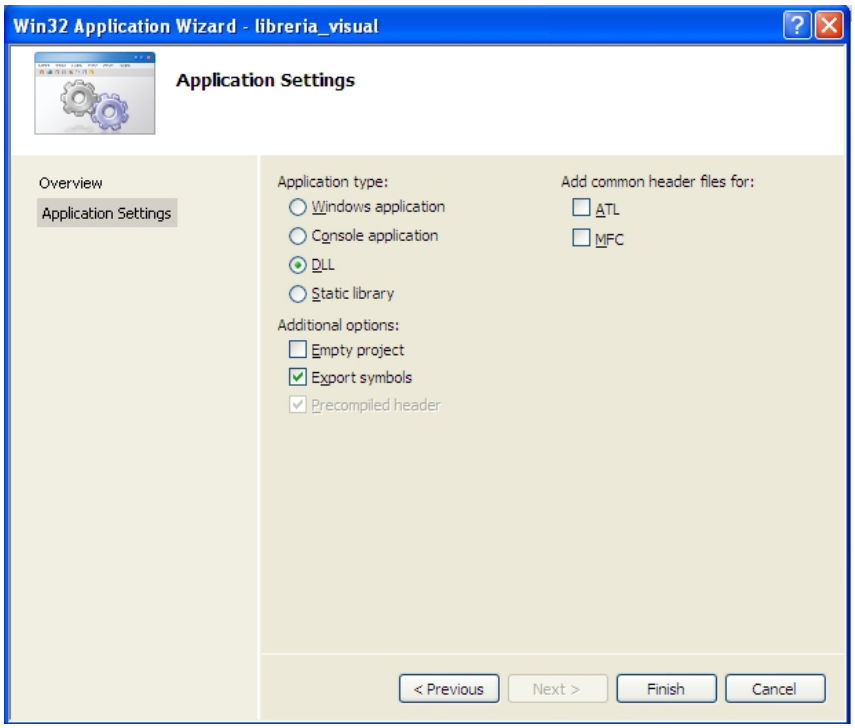


Figura 90. Configuración de la aplicación a tipo "DLL"

7. Presionar “Finish”.

8. Ahora tendremos un espacio de trabajo o “Solution Explorer” llamado “libreria_visual” donde nos se encuentran los siguientes archivos.
Véase figura 91.

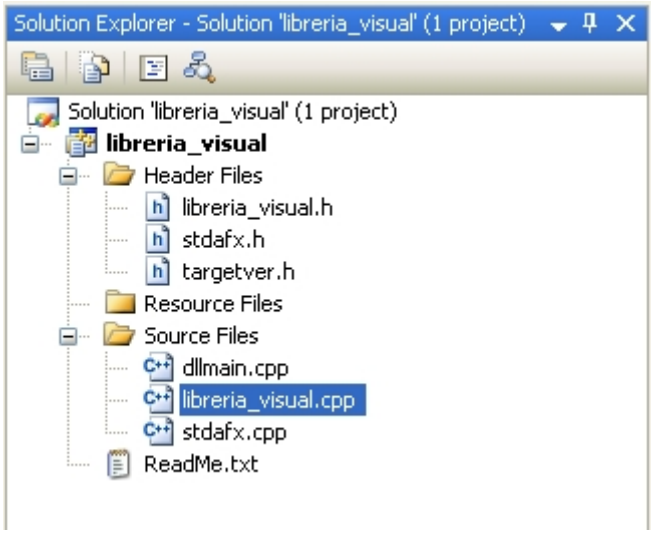


Figura 91. Visualización del espacio de trabajo de Visual Studio

El archivo "libreria_visual.h" es nuestro archivo de cabecera de la función "wrapper" y contiene las declaraciones de las funciones, variables de entrada y salida y clases que se van a exportar o importar.

El archivo "libreria_visual.cpp" es nuestro archivo fuente y contiene la estructura de la función "wrapper" donde se va a desarrollar el código para hacer accesible las funciones, variables de entrada y salida y las clases que se van exportar o importar.

9. Visual Studio ha creado una función "wrapper" que exporta una variable, una función y una clase. Sólo se usará la parte que exporta una función y se creará una función "wrapper" para exportar tres funciones.

10. En el archivo de cabecera "libreria_visual.h" se borrarán los ejemplos de declaración de funciones que se han generado automáticamente por el Visual Studio y se sustituirá por la declaración de las siguientes funciones. Véase figura 92.

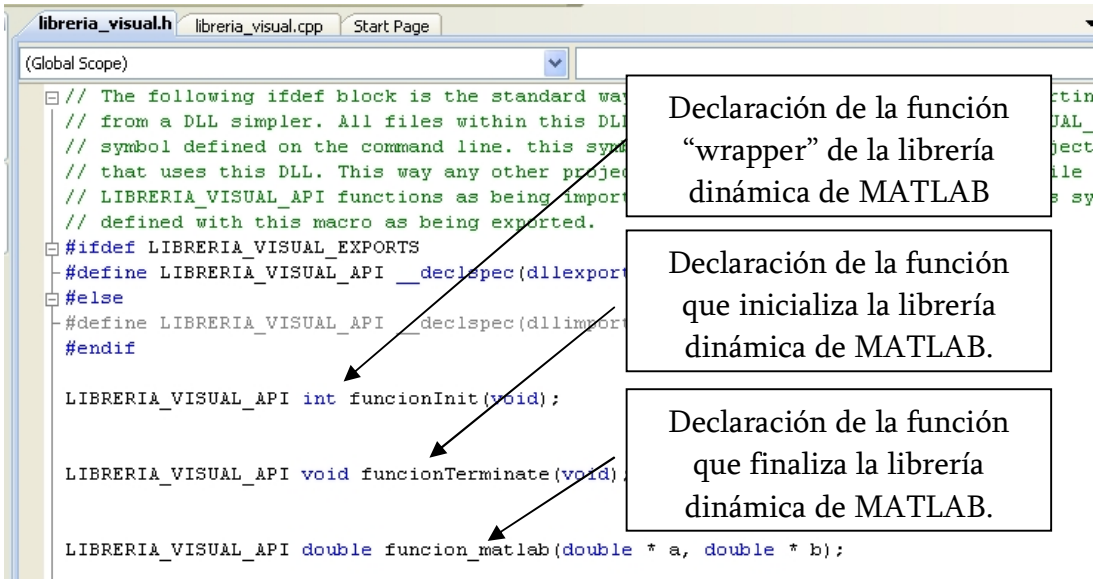


Figura 92. Declaración de las funciones de cabecera de la función "wrapper"

11. En el archivo fuente "libreria_visual.cpp" se borran los ejemplos de exportación de variables, funciones y clases que se han generado automáticamente por el Visual Studio y se sustituye por la estructura de la función "wrapper" que permite la importación y exportación de variables y funciones. Además se añade las inclusiones de servidor o "includes" que permitirán copiar el contenido de un fichero en otro fichero. La ventaja de las inclusiones es que permite agrupar código en ficheros y utilizarlo en todas las páginas que se quiera. Véase figura 93.

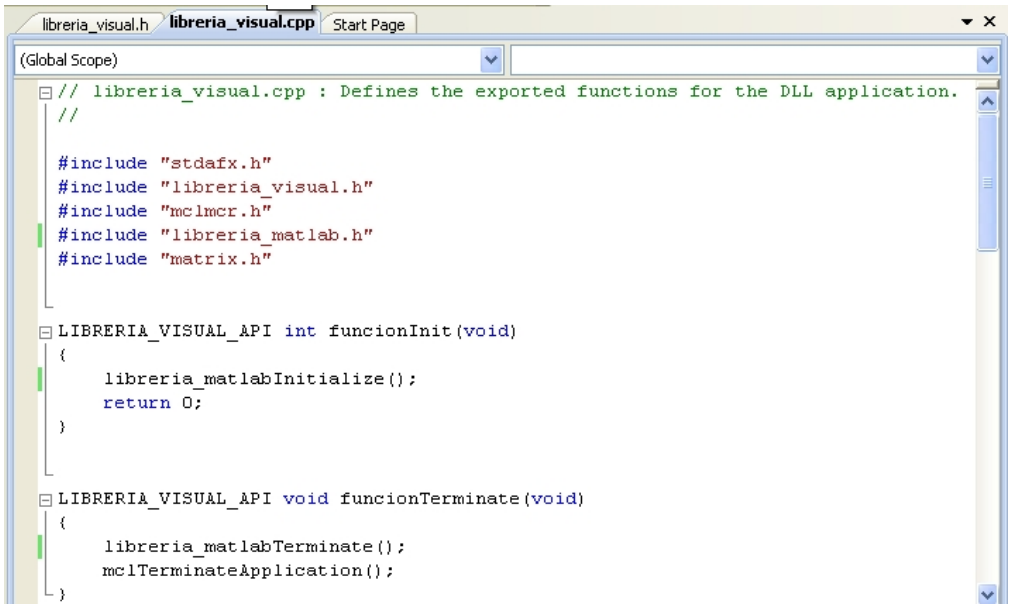


Figura 93. Declaración de las funciones de la función "wrapper" y declaración de "includes"

Los "mxArrays" son los únicos objetos con los que trabaja MATLAB. Todos los tipos de variables de MATLAB (escalares, vectores, matrices, cadenas de caracteres, estructuras, vectores de celdas, etc.) son "mxArrays". Por tanto, en la función "wrapper" deben hacerse referencia a estas variables con dicho formato.

Se acaba de crear la estructura para la función "wrapper" de la librería dinámica. Véase figura 93.

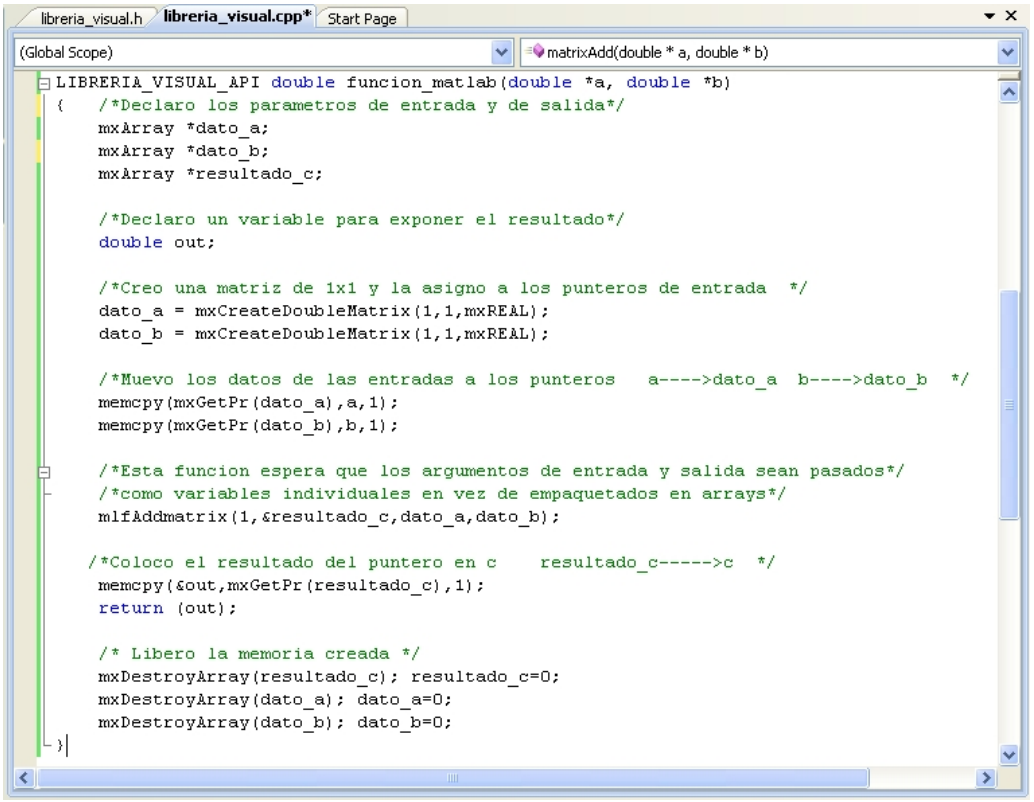


Figura 94. Estructura de la función "wrapper" de la librería dinámica

12. Build → Build Solution

A continuación, se define y configura el Microsoft Visual Studio para que la compilación y el link de la librería dinámica que se va a generar se ejecuten adecuadamente.

13. Clic derecho en "libreria_visual" → Add → Existing Item…Véase figura 95.

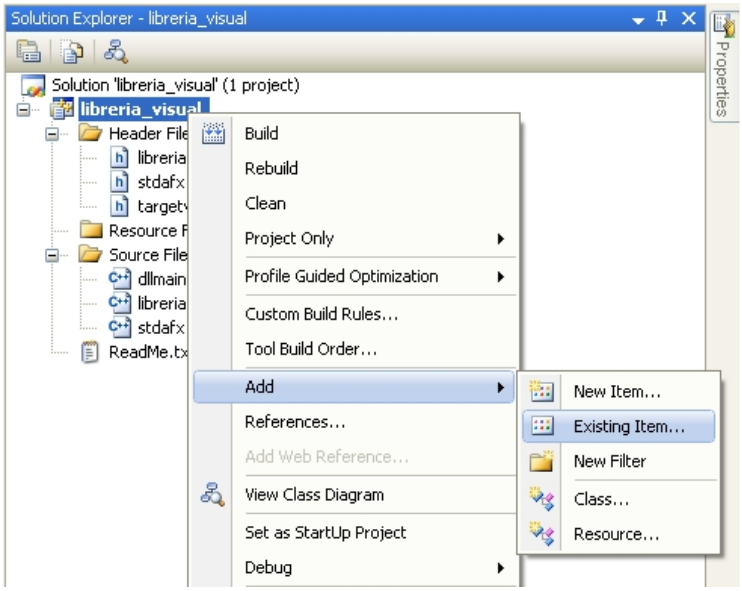


Figura 95. Añadir elementos existentes al proyecto

Se añade el archivo "libreria_matlab.lib" que se encuentra en la carpeta donde hemos compilado y generado la librería dinámica de MATLAB.

Se añade el archivo "mclmcrrt.lib" que se encuentra en la ruta: C:\Archivos de programa\MATLAB\R2008a\extern\lib\win32\microsoft\mclmcrrt.lib

Si no ha habido ningún problema, estos archivos **.lib* aparecerán en nuestra proyecto jerárquico. Véase figura 96.

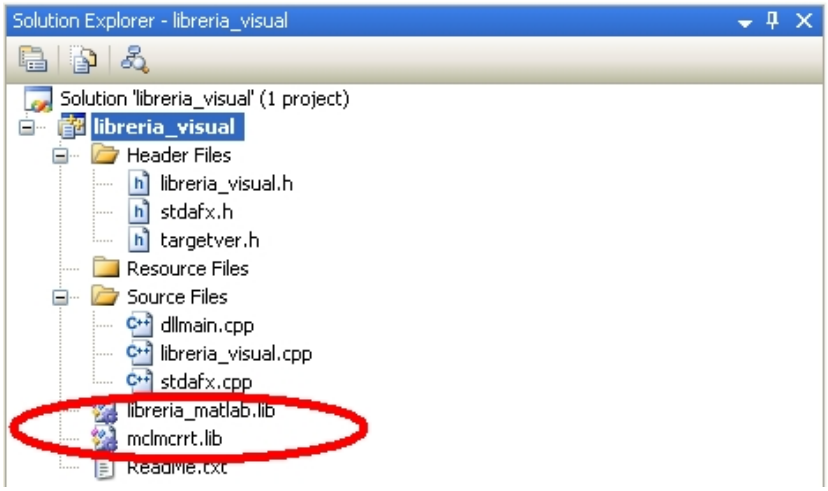


Figura 96. Visualización de los ficheros añadidos al proyecto

14. Clic derecho en "libreria_visual" → Propierties

- C/C++→ Code Generation → Runtime Library → Multi-threaded DLL (/MD)

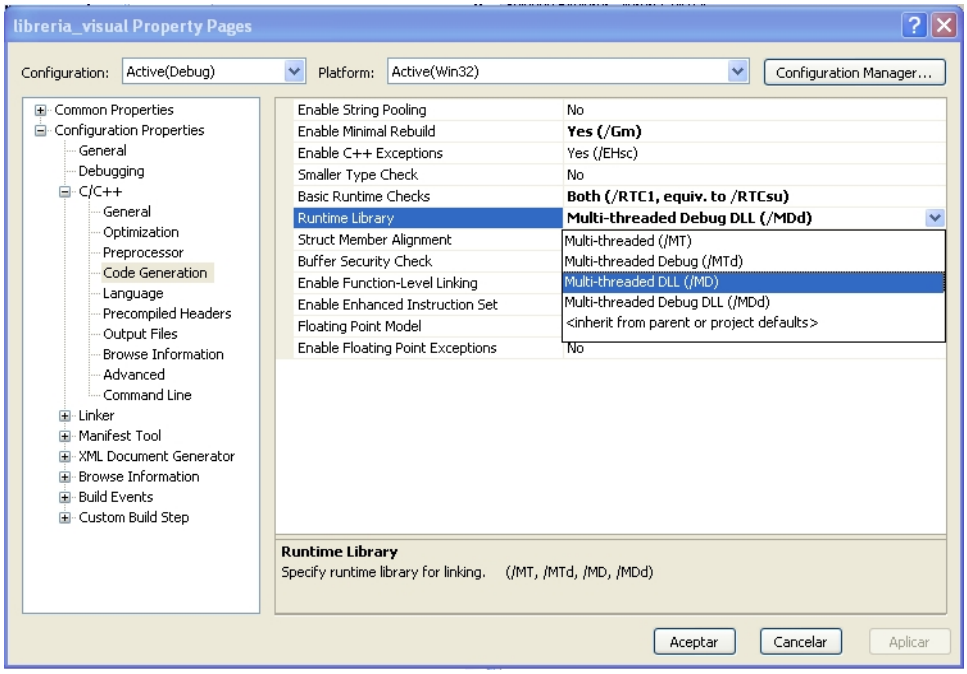


Figura 97. Seleccionar opciones del menú

- C/C++ → General → Additional Include Directories

Se incluye el directorio donde se encuentra "libreria_visual.h" y donde hemos generado la librería dinámica de MATLAB. Además se incluye el siguiente directorio: C:\Archivos de programa\MATLAB\R2008a\extern\include

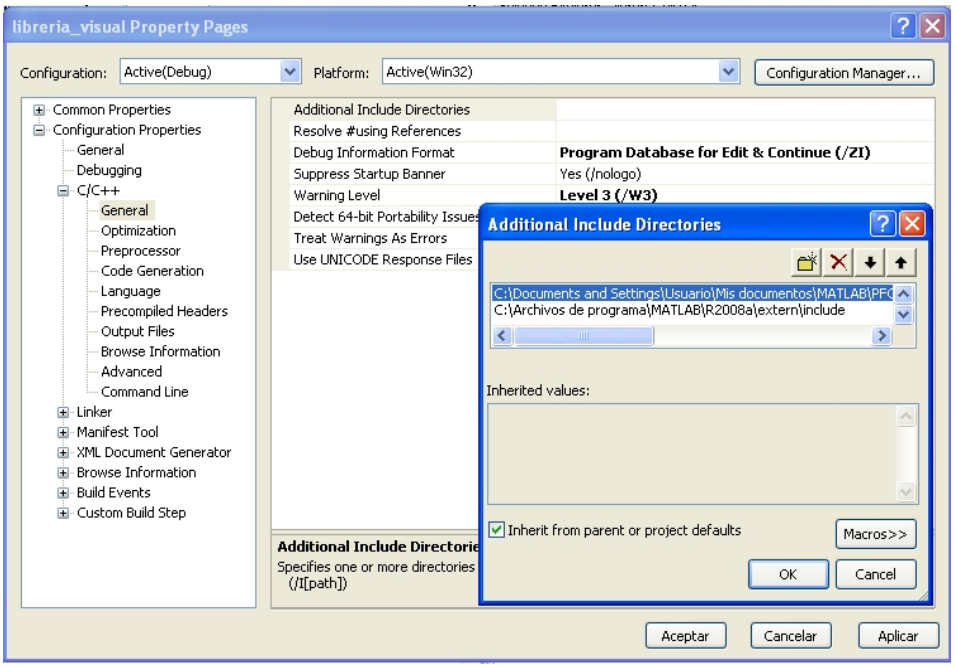


Figura 98. Añadir directorios adicionales

15. Esto debería compilarse sin ningún error y debería generar la nueva librería dinámica con la que podremos comunicar nuestra interfaz gráfica con aplicaciones externas.

    Si la compilación es satisfactoria, se incluirá una nueva carpeta llamada "Debug" dentro de nuestra carpeta de proyecto "libreria_visual" y a su vez, dentro de esta carpeta "Debug" se generará la nueva librería dinámica "librería_visual.dll".

16. Se coloca la librería dinámica generada por MATLAB "libreria_matlab.dll" en la misma carpeta "Debug" donde se ha generado la nueva librería dinámica "libreria_visual.dll".

Tras seguir todos los pasos anteriores, se consigue generar una librería dinámica programada en C++ capaz de intercambiar variables entre el programa LabVIEW y MATLAB.

3.5.1.2 MATLAB Script Node

Esta es la alternativa finalmente elegida para realizar la comunicación entre LabVIEW y MATLAB.

MATLAB Script Node es una función perteneciente al espacio de trabajo del programa LabVIEW que crea un nodo o bloque y en su interior se genera un MATLAB script, es decir, una hoja de trabajo de MATLAB.

Este sistema ofrece grandes posibilidades ya que al nodo generado se le permite añadir la ruta específica de la carpeta del proyecto de MATLAB para convertirse virtualmente en un nuevo fichero perteneciente a la carpeta de trabajo.

Además, se pueden añadir variables de entrada al nodo procedentes de otros programas generados en LabVIEW permitiendo interactuar dichas entradas con los ficheros **.m* establecidos en la carpeta de trabajo de MATLAB.

Del mismo modo, las variables de salida generadas por el software de MATLAB pueden ser transmitidas a una hoja de trabajo de LabVIEW concluyendo de esta manera una comunicación bidireccional de intercambio de variables entre ambos programas.

MATLAB Script Node es generalmente compatible con la sintaxis de todos los ficheros **.m* generados por MATLAB aunque la compatibilidad no es completa ya que hay algunas funciones y algoritmos que no están implementados.

La programación de código MATLAB que se ha realizado en el interior del MATLAB Script Node para comunicar ambos software, ha sido realizada mediante la creación de varios ficheros de intercambio de variables propios de MATLAB, que permiten por razón de los comandos “save” y “load” escribir y cargar las variables almacenadas en ellos.

Por último, cabe destacar, que este sistema de comunicación entre ambos programas es la manera más fácil e intuitiva de todas las alternativas, ya que se puede trabajar desde LabVIEW con la estructura y programación básica de MATLAB.

Seguidamente se van a describir algunas de las ventajas que han influido en la decisión por decantarse por el método de comunicación establecido:

- La primera ventaja de usar LabVIEW es que es compatible con herramientas de desarrollo similares y puede trabajara la vez con programas de otra área de aplicación, como MATLAB o Excel. Además se puede utilizar en muchos sistemas operativos, incluyendo Windows y UNIX, siendo el código transportable de uno a otro.

- Otra de las ventajas más importantes que tiene este lenguaje de programación es que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes).

- Es muy simple de manejar, debido a que está basado en un nuevo sistema de programación gráfica, llamado lenguaje G.

- Es un programa enfocado hacia la instrumentación virtual, por lo que cuenta con numerosas herramientas de presentación, en gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y versátiles. Por tanto, se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.

- Es un programa que contiene librerías especializadas para manejos de DAQ (tarjetas de adquisición de datos), Redes, Comunicaciones, Análisis Estadístico, Comunicación con Bases de Datos (útil para una automatización de una empresa a nivel total).

- Como se programa creando subrutinas en módulos de bloques, se pueden usar otros bloques creados anteriormente como aplicaciones por otras personas.

- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.

- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.

En el ambiente de trabajo de LabVIEW existen dos paneles, el panel frontal y el panel de programación ó diagrama de bloques. En el panel frontal se diseña la interfaz de usuario y en el panel de programación se relacionan los elementos utilizados en la interfaz mediante operaciones que determinan cómo funciona el programa o el sistema.

- Panel Frontal

  Se trata de la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc.

  Cada uno de ellos puede estar definido como un control o un indicador. Los controles sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación.

- **Diagrama de bloques**

  En cambio, en el diagrama de bloques se aprecia la estructura del programa, su función y algoritmo de forma gráfica en lenguaje G, en el cual los datos fluyen a través de líneas de conexión conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.

  En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado de las entradas y salidas que se crearon en el panel frontal.

  Al diseñar el programa de forma gráfica, se hace visible una programación orientada al flujo de datos, donde se tiene una interpretación de los datos también de forma gráfica.

  El flujo de datos va de izquierda a derecha en el panel de programación y esta determinado por las operaciones o funciones que procesan los datos. Es fácil observar en el panel de programación como se computan los datos en cada parte del programa cuando se realiza una ejecución del programa paso a paso. En LabVIEW las variables se representan mediante una figura tanto en el panel frontal como en el panel de programación, de esta forma se puede observar su respuesta en la interfaz del usuario y en el flujo de datos del código del programa.

A continuación, va explicarse mediante una serie de pasos la programación gráfica que se ha realizado para la poderse llevar a cabo la comunicación entre los programas MATLAB y LabVIEW.

Para ello, se ha utilizado el nodo de "MATLAB Script" que permite relacionar ambos programas, gracias a la especificación de la ruta donde se encuentra el directorio de trabajo de MATLAB.

1. Ejecutar el programa LabVIEW.

2. Seleccionar Blank VI.

   El programa abrirá dos paneles, el Panel Frontal y Diagrama de Bloques.

3. Seleccionar el panel Diagrama de Bloques y pulsar el botón secundario para elegir la función "MATLAB Script" mediante la siguiente secuencia de comandos. Véase figura 98.

Mathematics → Scripts & Formulas → Script Nodes → MATLAB script

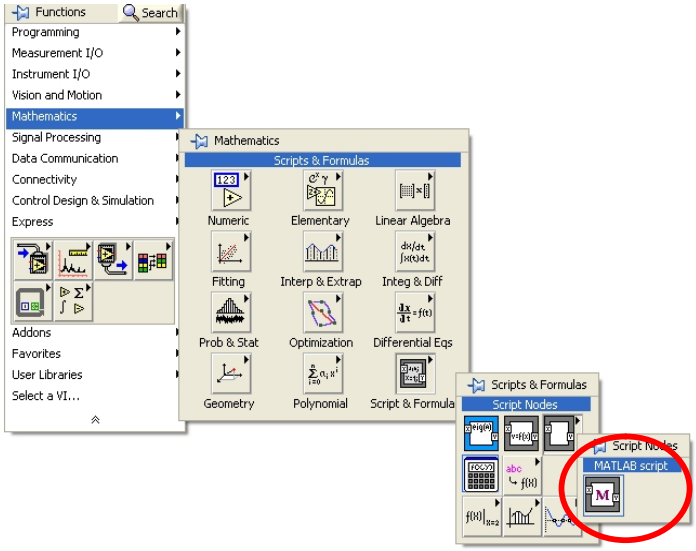


Figura 99. Selección del nodo " MATLAB Script "

4. Dibujar un hoja de trabajo con dimensiones suficientes para realizar la programación del “MATLAB script”.

5. Seleccionar los siguientes componentes para formar el “path”, la ruta donde se encuentra la carpeta de trabajo de MATLAB. Véase figura 100, 101 y 102.
Programming → File I/O → Build Path

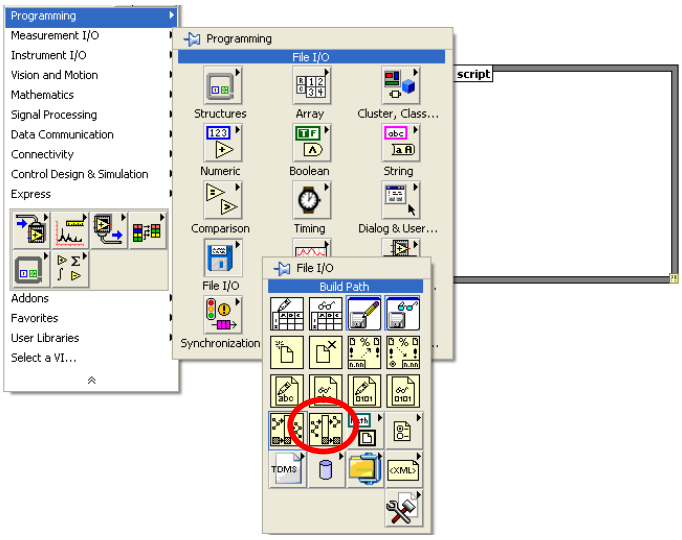


Figura 100. Seleccionar "Build Path"

Programming → File I/O → File Constants → Path Constant

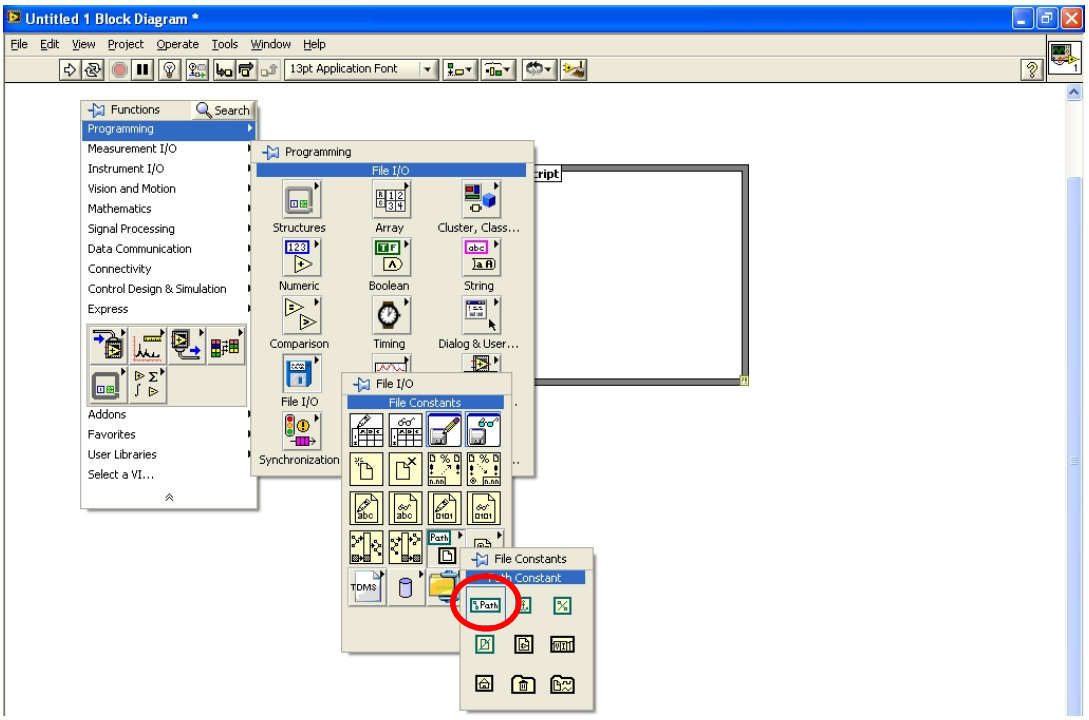


Figura 101. Seleccionar "Paht Constant"

Programming → File I/O → File Constants → Default Directory

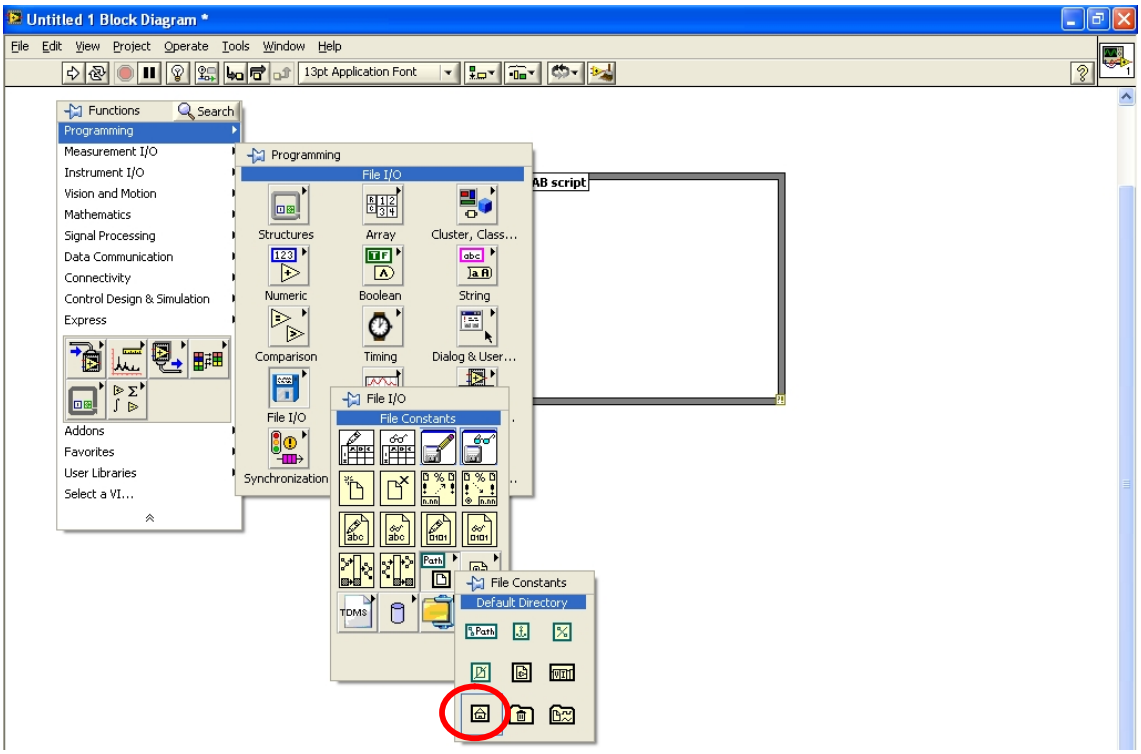


Figura 102. Seleccionar "Default Directory"

6. Nombrar la carpeta específica de trabajo de MATLAB en el componente "Path Constant" y especificar la ruta donde se encuentra dicha carpeta para el componente "Default Directory". Véase figura 104.

   Para especificar la ruta donde se encuentra la carpeta de trabajo seleccionar: Tools → Options…

   A continuación, seleccionar la categoría "Paths", elegir la opción "Default Directory" e introducir la ruta específica utilizando el comando "Browse…". Por último pulsar el botón "Replace" y reiniciar LabVIEW para que surjan efecto los cambios establecidos. Véase figura 103.

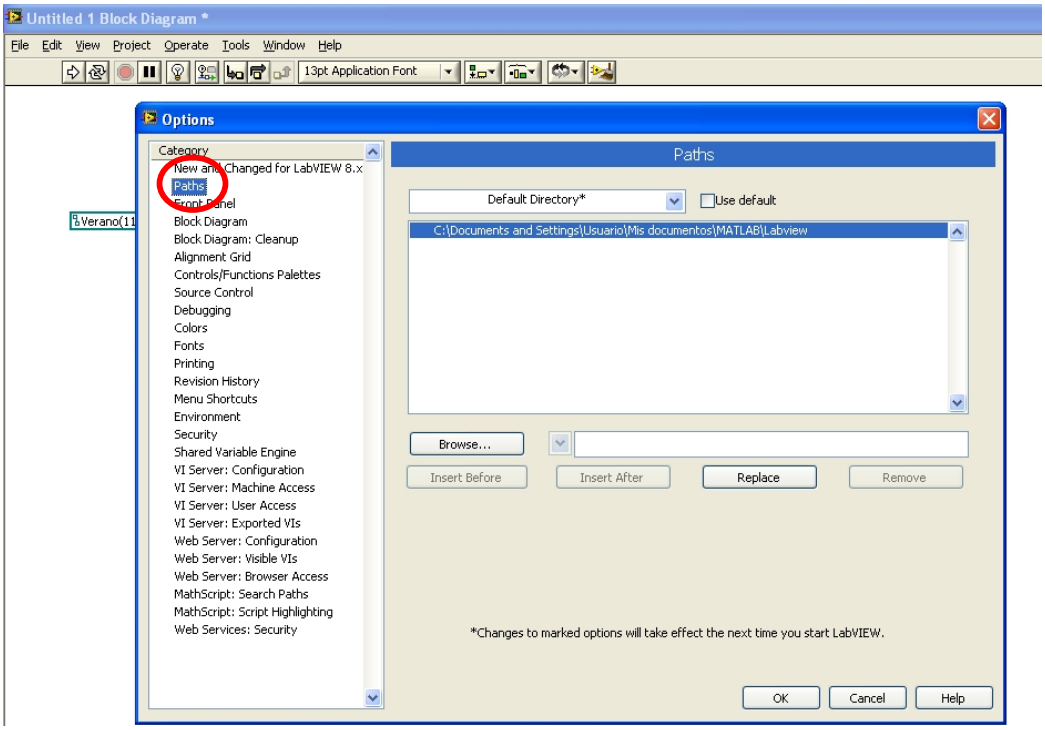


Figura 103. Selecciona categoría "Paths" y elegir la ruta para el "Default Directory"

7. Realizar las conexiones de los componentes y añadir una entrada de datos cambiándola a tipo "Path" para poder introducir la ruta completa de la carpeta de trabajo de MATLAB al "MATLAB script". Véase figura 104.

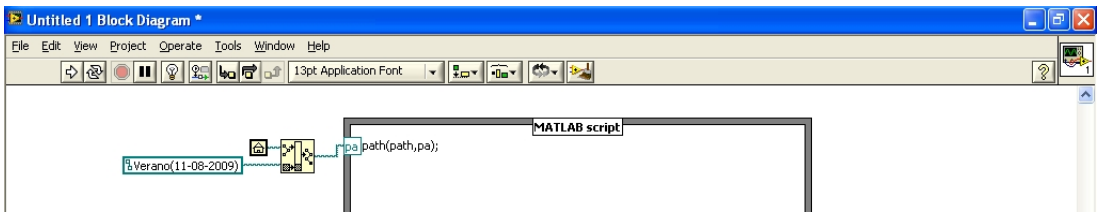


Figura 104. Conexión de elementos para formar la ruta del nodo de "MATLAB Script"

Finalmente, elegir un nombre para la nueva entrada y escribir el comando en el "MATLAB script" para que cuando se ejecute el programa lea la ruta especificada. Véase figura 108.

8. Introducir las entradas procedentes del LabVIEW e introducir las salidas generadas por MATLAB.

Para introducir las entradas se pulsa el botón secundario y se elige la opción "Add Input". Acto seguido, se pulsa el botón secundario sobre la etiqueta generada y se elige la opción "Create → Control" para generar un control numérico que esté referenciado por medio de la etiqueta a las variables de entrada de LabVIEW. Véase figura 105.

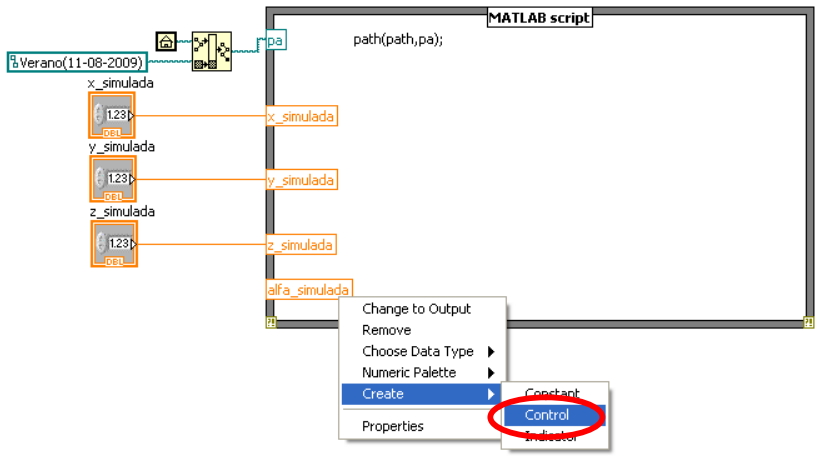


Figura 105. Crear controles para las entradas añadidas

Del mismo modo, para introducir las salidas se pulsa el botón secundario y se elige la opción "Add Output". A continuación, se pulsa el botón secundario sobre la etiqueta generada y se elige la opción "Create → Indicator" para generar un indicador con su correspondiente etiqueta para referenciar dichas variables de salida. Véase figura 106.

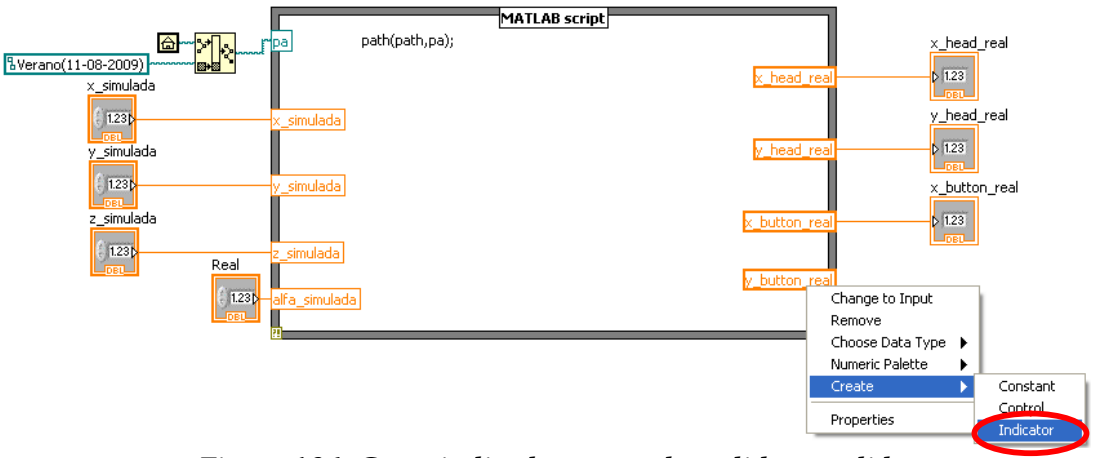


Figura 106. Crear indicadores para las salidas añadidas

9. Programar la función "MATLAB script" para que guarde las variables de entrada procedentes de LabVIEW en un archivo de intercambio de datos y cargue de dicho archivo de intercambio de datos las salidas procedentes de MATLAB. Véase figura 107.

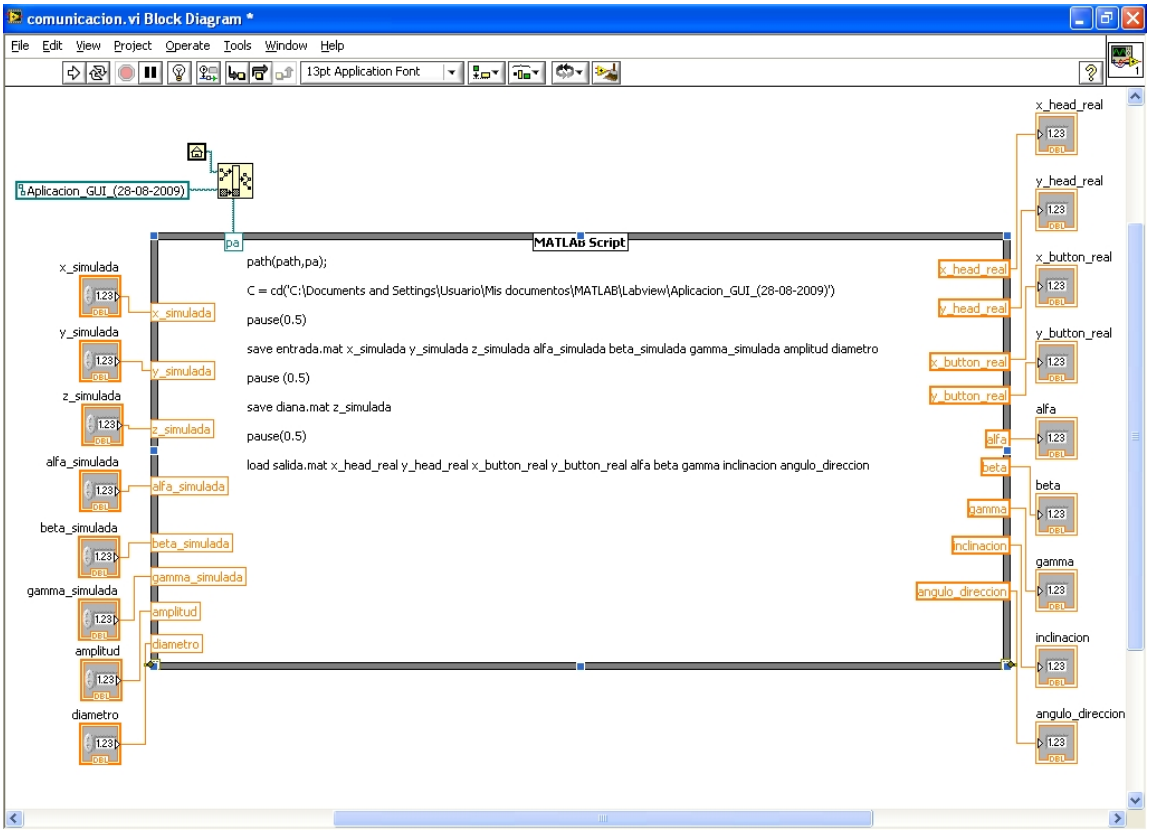


Figura 107. Esquema final de las conexiones del nodo de "MATLAB Script"

Tras finalizar todos estos pasos, tenemos programado el diagrama de bloques, donde se aprecian todas las conexiones de las entradas y salidas para realizar el flujo de datos. A continuación, se muestra el panel frontal donde se reflejan todos los indicadores visuales para comprobar los valores numéricos. Véase figura 108.

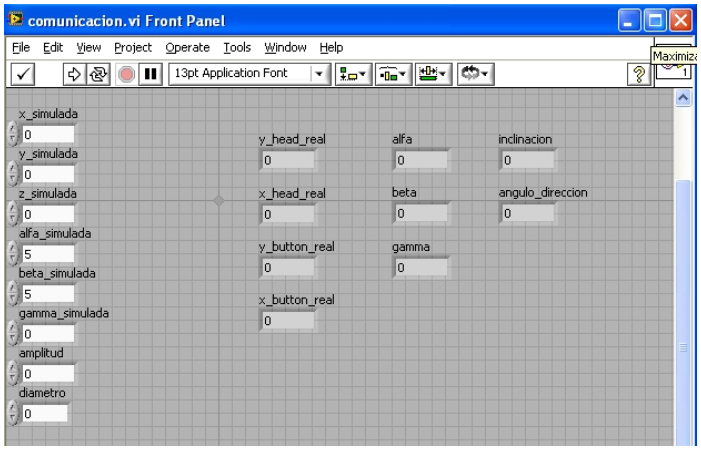


Figura 108. Panel frontal de las conexiones del nodo de "MATLAB Script"

3.5.2 Alternativas de comunicación entre aplicaciones MATLAB

3.5.2.1 Variables globales

Para realizar una comunicación entre dos programas diferentes de MATLAB mediante variables globales, en primera instancia puede parecer una opción sencilla y accesible.

Las variables definidas dentro de una función son variables locales, es decir, las variables que se crean dentro de la función pertenecen al espacio de trabajo de la función y son inaccesibles desde otras partes del programa. Además no interfieren con variables del mismo nombre definidas en otras funciones o partes del programa y desaparecen una vez finaliza la llamada a la función.

Por tanto, para que la función se pueda comunicar y tener acceso a variables que no han sido pasadas como argumentos es necesario declarar dichas variables como variables globales, tanto en el programa principal como en las distintas funciones que deben acceder a su valor. Efectivamente, su valor es visible en todos los espacios de trabajo de las funciones donde han sido declaradas. Es decir, el ámbito de una variable global son todas las funciones que componen el programa, cualquier función puede acceder a dichas variables para leer y escribir en ellas y además se puede hacer referencia a su dirección de memoria en cualquier parte del programa.

Sin embargo, este método no es válido para realizar una comunicación de variables entre diferentes programas. La razón por la cual no se puede llevar a cabo la comunicación es porque al crear los ejecutables externos (**.exe*) de cada uno de los programas que se quieren comunicar, el espacio de trabajo donde se almacenan todas las variables globales deja de ser común a ambos programas al trabajar de forma externa a MATLAB.

Por ese motivo, trabajar desde el ámbito de MATLAB para realizar intercambios de variables resulta sencillo y cómodo porque las variables globales están al alcance en cualquier momento, pero para conseguir realizar una comunicación mediante aplicaciones externas es necesario recurrir a otros métodos más certeros.

3.5.2.2 Fichero de intercambio de datos (.mat)*

Este es el método finalmente elegido para realizar la comunicación, se basa en la creación y uso de un fichero compartido que permita leer y escribir tanto el nombre de la variable como su valor, siendo accesible en cualquier momento por cualquiera de los tres software que están comunicados entre ellos.

No presenta el problema de tener varios programas en diferentes carpetas de trabajo, puesto que se puede hacer la llamada al archivo indicando previamente la ruta o "path" donde está ubicado, beneficiando la independencia de cada programa.

Este tipo de archivo con la extensión **.mat*, no sólo tiene comandos específicos para su escritura y lectura, sino que además mantiene el formato que tengan las variables. Esto beneficia enormemente la programación del software ya que no es necesario tener que especificar el tipo de formato junto al valor a la hora de escribirlo o leerlo. Estas ventajas no se obtienen en otros tipos de archivos comúnmente utilizados, como los archivos de texto plano **.txt* o de tabla **.xls.*

Por todos estos motivos, se ha elegido dicho método para realizar la comunicación.

A continuación, se va describir brevemente los comandos utilizados para realizar la comunicación entre los tres software y su funcionalidad.

El comando "save" permite guardar el valor de todas las variables que se deseen almacenar en el fichero **.mat*, tan sólo especificando los nombres de las variables consecutivamente. Además, mediante la instrucción "-append" al final del comando "save", crea variables nuevas en el archivo en caso de que no hayan sido almacenadas anteriormente o sobrescribe su valor si ya habían sido almacenadas. El uso de la instrucción "-append" evita que se borren otras variables guardadas en el archivo de intercambio ya que no todos los software harán uso de las mismas variables del archivo.

```
save datos.mat x y alfa beta gamma acceso -append
```

El comando "load" lee el valor de las variables y las guarda en el archivo de intercambio con el mismo nombre si están siendo usadas por el programa. Sin embargo, las genera nuevas si no han sido declaradas todavía. Al igual que el comando "save", nos permite hacer una lectura selectiva de las variables especificando el nombre de las variables elegidas consecutivamente, sin la necesidad de tener que cargar todas las variables, ocupando espacio en memoria innecesario y evitando posibles pérdidas de valores.

```
load datos.mat z_simulada
```

Por tanto, para la comunicación de variables, vectores o matrices entre dos ficheros **.m* ya sean funciones o scripts, se realiza mediante la creación de un fichero común con extensión **.mat*.

Una dificultad añadida a la hora de compartir un mismo archivo para guardar las variables de intercambio, es la necesidad de regular el tráfico de los programas al acceder por medio de lectura o escritura al archivo. En el caso de la comunicación Diana – GUI, se ha tomado una variable común llamada "acceso", la cual es puesta por la diana a valor lógico "1" cuando se realiza algún nuevo cálculo de la situación de la tuneladora y, al ser leída por la GUI le indicará que puede realizar la representación de una nueva posición de la tuneladora.

En cuanto a la comunicación MATLAB – LabVIEW, que en principio puede parecer imposible por no poder usar este tipo de archivos en LabVIEW, se ha podido llevar a cabo gracias a la inclusión en la últimas versiones de LabVIEW del bloque de programación llamado "MATLAB Script Node", con el cual se puede insertar directamente código fuente de MATLAB siendo reconocido y ejecutado, por lo que el método llevado en LabVIEW es exactamente igual que en los programas de la Diana y GUI realizados en MATLAB.

Por último, indicar que se han creado hasta cuatro archivos ".mat" para conseguir la total comunicación de los programas. La finalidad de utilizar varios archivos para la comunicación, es evitar que varios programas realicen a la vez la escritura o lectura de un mismo archivo, lo que llevaría a un error por parte de uno o varios programas por imposibilidad de obtener o guardar los datos. Los archivos son: "salida.mat", "entrada.mat", "diana.mat" y "gui.mat".

3.5.2.3 Socket

La comunicación por Socket es comúnmente usada para realizar el intercambio de datos en una red Ethernet bajo el tipo cliente-servidor. Más exactamente, un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información.

Este tipo de comunicación necesita ser configurado mediante una dirección IP, un protocolo de transporte y un número de puerto, por lo que está muy orientado a la comunicación para programas que se ejecuten en diferentes máquinas. En el caso de nuestro sistema no es necesario, puesto que se ejecutarán todos los programas en el mismo ordenador, por lo que no es a priori una ventaja frente a otros métodos puesto que además obliga a realizar la configuración de los puertos de red.

El inconveniente que supone tener que dar formato a los valores obtenidos en esta comunicación, en función del dato que se vaya a recibir, ha sido fundamental a la hora de desestimar este método de comunicación frente al elegido.

Sin embargo, hay que tener presente este método de comunicación en el caso de que se realizase la integración de un ordenador industrial en la propia diana, con lo que tendría muchas posibilidades de ser el más adecuado, teniendo en cuenta que se realizaría la transmisión de datos desde la diana al ordenador central por medio de conexión de red.

4. Resultados obtenidos

4.1 Manual de uso

4.1.1 Inicialización de la interfaz gráfica externa

Para iniciar el programa que ejecuta la interfaz gráfica externa se puede inicializar desde el propio programa MATLAB o se puede generar un ejecutable para lanzar la interfaz gráfica sin la necesidad de tener instalado el programa completo.

En primer lugar, para iniciar la interfaz gráfica desde MATLAB hay que seguir los pasos detallados a continuación.

1. Ejecutar MATLAB.

2. Abrir en el "Current Directory" la carpeta específica de trabajo del software de la interfaz gráfica.

3. Escribir en el "Command Window" el nombre del fichero que inicializa el sistema sin la extensión **.m* y pulsar *ENTER*.

En segundo lugar, para crear una aplicación ejecutable (Standalone application) hay que seguir una serie de pasos.

1. Ejecutar MATLAB.

2. Abrir en el "Current Directory" la carpeta específica de trabajo del software de la interfaz gráfica.

3. Escribir en el "Command Window": *mcc -m presentacion.m* y pulsar *ENTER* para generar la aplicación ejecutable (Standalone application) en el directorio actual de la carpeta específica de trabajo dentro del "Current Directory".

4. Para ejecutar la aplicación que se acaba de generar hay dos métodos posibles.

   En primer lugar, ejecutar directamente la aplicación ejecutable desde la propia carpeta de trabajo donde se ha generado pulsando sobre ella doble clic ó pulsando botón secundario y eligiendo la opción "Open".

   En segundo lugar, ejecutar el "Command Prompt" y especificar la ruta completa donde se encuentra la aplicación ejecutable y escribir el nombre del archivo con la terminación **.exe*.

A continuación, se muestra la interfaz de guiado en pleno proceso de monitorización de la trayectoria de la máquina tuneladora en dos dimensiones. Ver figura 109.

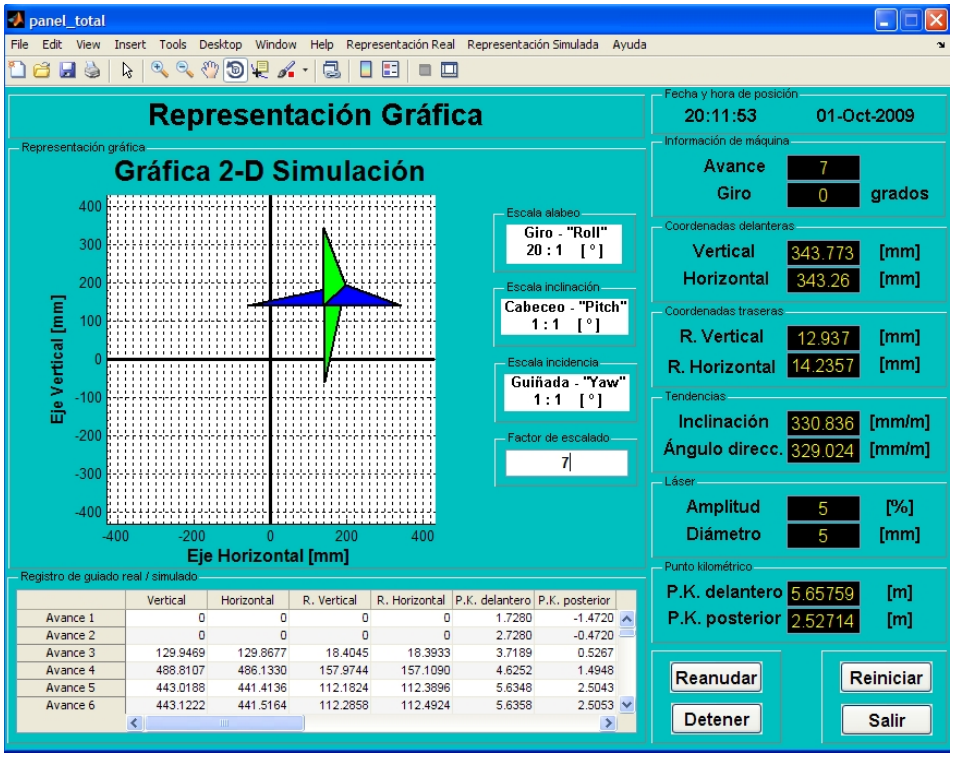


Figura 109. Interfaz gráfica de guiado en modo representación en dos dimensiones

A continuación, se muestra la interfaz de guiado en pleno proceso de monitorización de la trayectoria de la máquina tuneladora en tres dimensiones. Ver figura 110.

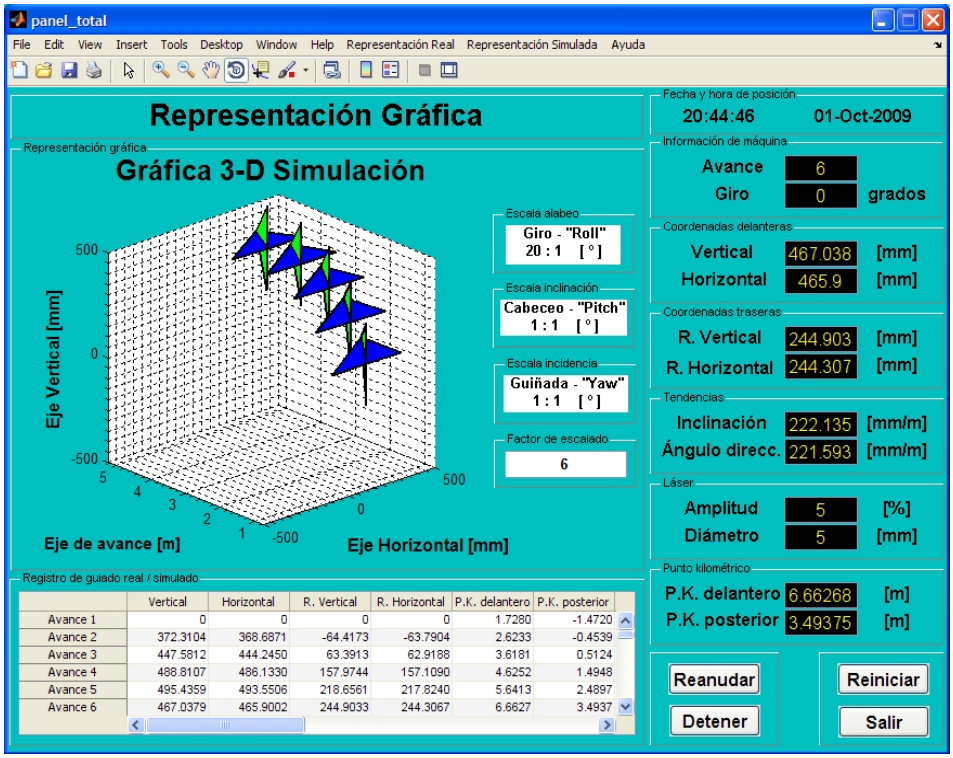


Figura 110. Interfaz gráfica de guiado en modo representación en tres dimensiones

4.1.2 Funcionamiento de la interfaz gráfica externa

Conceptos previos antes de iniciar la monitorización de la interfaz gráfica

1. La máquina tuneladora avanza siempre por donde le sea más fácil, normalmente cogiendo una tendencia a elevarse. El tipo de terreno va a facilitar o empeorar el guiado de la máquina. Los terrenos no cohesivos son problemáticos y dificultan el control de las maniobras. Por lo tanto, es fundamental, que el usuario realice las correcciones oportunas con suficiente antelación para no desviarse excesivamente de la referencia.

2. Una vez que se haya inicializado el sistema de guiado y se ejecute la interfaz gráfica externa, lo primero que se debe hacer es elegir el factor de escalado de la máquina tuneladora para que se produzcan las representaciones gráficas con una escala mayor o menor. El factor de escalado hace una reducción proporcional de las dimensiones de la máquina, tantas veces como el número que se le indique en el editor de texto. Por tanto, si se tiene activado un factor de escalado de 1, se estarán representando las dimensiones reales de la máquina tuneladora. Sin embargo, si se tiene activado un factor de escalado de 6, se estarán representando las dimensiones de la máquina tuneladora seis veces menor a sus dimensiones reales.

3. La monitorización de representaciones gráficas en dos y tres dimensiones no son reales con respecto a las dimensiones totales de la máquina tuneladora sino se utiliza un factor de escalado de 1. Para el resto de los casos, se realiza un escalado reductor de la flecha simbólica que tipifica la máquina tuneladora. Ésta suele seguir una trayectoria bastante rectilínea, es decir, se producen pocas desviaciones y cambios de trayectoria. Por ese motivo, se utiliza un factor de escalado entre 6 y 10, ya que se amplifican dichas desviaciones y cambios de trayectoria para que el usuario que controla la interfaz gráfica las visualice más nítidamente y pueda aplicar con antelación las correcciones oportunas en el guiado de la máquina tuneladora.

4. Los giros que se producen sobre el eje longitudinal de la máquina y se reflejan en la interfaz gráfica, tienen aplicado un escalado el cual, 1° en la realidad equivale a 20° en la interfaz gráfica. Esto es debido a que la máquina tuneladora sólo gira sobre dicho eje cuando se produce un atasco en el cabezal cortante y empieza a girar la máquina sobre sí misma. Por tanto, para no producir grandes averías por el retorcimiento de los cables internos, se amplía mucho este giro para visualizar rápidamente en la interfaz gráfica el problema y detener la máquina tuneladora.

Descripción todos los valores paramétricos que monitoriza la interfaz gráfica

- **Vertical [mm]:** coordenada de la parte delantera de la máquina tuneladora que indica el desplazamiento vertical respecto al cero de la tarjeta.

- **Horizontal [mm]:** coordenada de la parte delantera de la máquina tuneladora que indica el desplazamiento horizontal respecto al cero de la tarjeta.

- **R. Vertical [mm]:** coordenada de la parte trasera de la máquina tuneladora que indica el desplazamiento vertical respecto al cero de la tarjeta.

- **R. Horizontal [mm]:** coordenada de la parte trasera de la máquina tuneladora que indica el desplazamiento horizontal respecto al cero de la tarjeta.

- **Inclinación [mm/m]:** valor que se obtiene del cálculo del valor absoluto de la diferencia entre las coordenadas verticales de la parte delantera y trasera de la máquina tuneladora. Este valor indica la tendencia vertical.

- **Ángulo direcc. [mm/m]:** valor que se obtiene del cálculo del valor absoluto de la diferencia entre las coordenadas horizontales de la parte delantera y trasera de la máquina tuneladora. Este valor indica la tendencia horizontal.

- **Giro [grados]:** valor que indica la cantidad de grados que ha girado la máquina sobre si misma. Esto giro es conocido como "Roll".

- **Amplitud del láser [%]:** Indica en que porcentaje ha variado la intensidad de la señal del haz inicial respecto al recibido en la tarjeta.

- **Diámetro del láser [mm]:** Indica el diámetro del haz láser recibido en la tarjeta a bordo de la máquina tuneladora.

- **Avance:** valor que indica el número de avances que se han producido durante todo el proceso de la tunelación.

- **P.K delantero [m]:** valor que indica la distancia recorrida desde el comienzo de la tunelación hasta la parte delantera de la posición actual donde se encuentre la máquina tuneladora.

- **P.K trasero [m]:** valor que indica la distancia recorrida desde el comienzo de la tunelación hasta la parte trasera de la posición actual donde se encuentre la tuneladora.

- **Fecha y hora de guiado:** indica la fecha y hora actualizada de cada avance y representación gráfica que se produce en la máquina tuneladora.

Funcionamiento y aplicación de los botones de la interfaz gráfica

El botón "Salir" se pulsa exclusivamente cuando se termina de tunelar todo el recorrido previsto por la máquina tuneladora. Una vez que se haya abandonado el panel de control de la interfaz gráfica se almacenan los registros de guiado en un archivo Excel y ya no se puede retomar la sesión con los datos anteriormente almacenados. Es decir, si se pulsa el botón "Salir" por equivocación y se vuelve a ejecutar el software, retomará la monitorización del proceso de tunelación partiendo de las condiciones iniciales y por tanto, se tendrán que reprogramar las coordenadas y sistemas de referencia a la nueva posición de la máquina tuneladora.

El botón "Reiniciar" se pulsa en el caso que se produzca alguna incidencia técnica y el sistema no se pueda recuperar. De este modo, si se produce algún fallo tanto en la representación gráfica como en la paramétrica, se pulsa el botón "Reiniciar" para que el sistema se reinicie evitando salir y volver a ejecutar la aplicación. Del mismo modo que el botón "Salir", una vez se halla reiniciado el sistema se tendrán que reprogramar las coordenadas y sistemas de referencia a la nueva posición donde se encuentre la máquina tuneladora.

El botón "Reiniciar" atenderá a su rutina de atención a la interrupción independientemente de que se encuentre detenida o en movimiento la monitorización de los parámetros y representaciones gráficas de la interfaz.

El botón "Detener" se pulsa en el caso que se produzca un fallo en el sistema y la máquina tuneladora necesite estar detenida durante el tiempo que dura la reparación. Una vez terminadas las reparaciones, se pulsa el botón "Reanudar" para proseguir en el estado en que se encontraba la máquina tuneladora antes de que se produjera la detención. Del mismo modo, se pueden utilizar estos botones para la finalización del día de trabajo y reanudación de la jornada siguiente.

Si por alguna circunstancia se pulsa el botón "Reanudar" mientras se está realizando el proceso de tunelización, el sistema permanecerá inalterado y la interfaz gráfica seguirá monitorizando sin incidencia alguna.

Así mismo, si el sistema se encuentra detenido por alguna incidencia técnica o por una simple pausa en el trabajo, si se pulsa el botón "Salir" para abandonar la aplicación, el programa responde a su rutina de interrupción, cierra la sesión y almacena en un archivo Excel los datos recibidos hasta dicho momento.

Del mismo modo, si el sistema se encuentra en pleno proceso de monitorización de parámetros y gráficas y se pulsa el botón "Salir", el software cierra la aplicación y guarda los datos almacenados hasta ese momento en un archivo Excel.

Funcionamiento y aplicación de los componentes de la paleta de herramientas

En la figura 111, se muestran todos los componentes de la paleta de herramientas integrada en la interfaz gráfica externa.

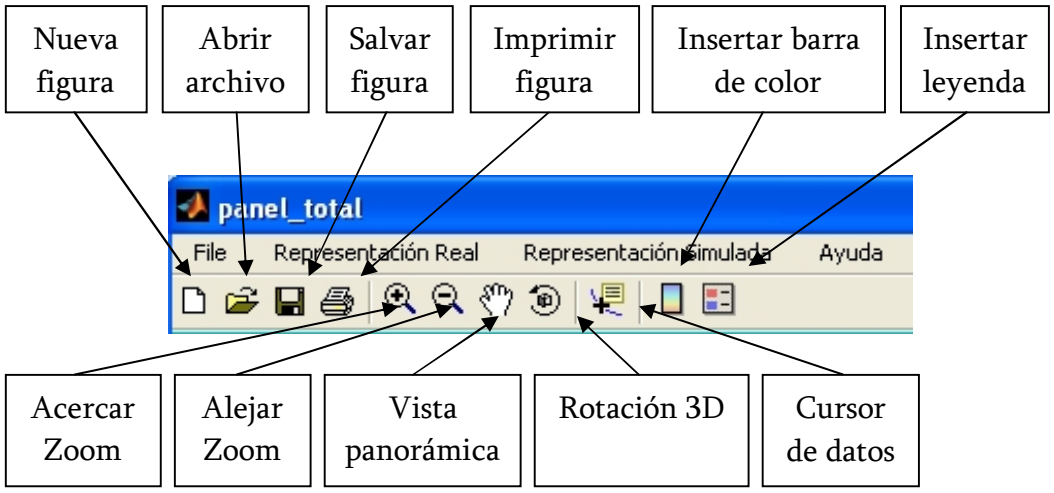


Figura 111. Paleta de herramientas de la interfaz gráfica externa

A continuación, se va describir la funcionalidad de los iconos más útiles para la monitorización del guiado de la máquina tuneladora.

1. **Rotación 3D**

Si se pulsa el botón de "Rotación 3D" se sustituye el puntero del ratón por una flecha rotativa que permite cambiar el ángulo de visión y la perspectiva de los ejes con tan sólo mantener pulsado el botón izquierdo del ratón y mover el puntero por cualquier punto del eje cartesiano.

Si se pulsa el segundo botón del ratón sobre los ejes se despliega un menú secundario que permite manejar algunas opciones interesantes. Véase figura 112.

- **Reset to Original View:** permite regresar a los ejes a su posición inicial de visión establecida.
- **Go to X-Y view:** permite posicionar a los ejes desde el punto de vista donde sólo se visualizan los ejes X-Y.
- **Go to X-Z view:** permite posicionar a los ejes desde el punto de vista donde sólo se visualizan los ejes X-Z.
- **Go to Y-Z view:** permite posicionar a los ejes desde el punto de vista donde sólo se visualizan los ejes Y-Z.

En el menú secundario "Rotate Options", es decir, opciones rotativas, se despliega un sub-menú con otras opciones:

- **Plot Box Rotate:** dibuja una caja alrededor de los ejes.
- **Continuous Rotate:** realiza una rotación de ejes continua.
- **Stretch-to-Fill Axes:** expande los ejes.
- **Fixed Aspect Ratio Axes:** fija los ejes en función de su orientación.

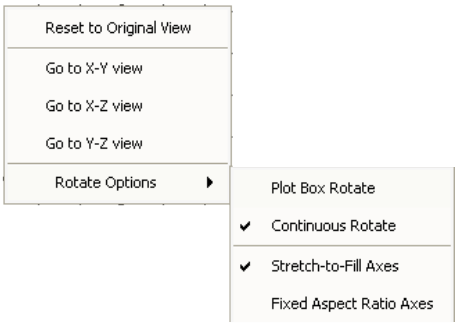


Figura 112. Menú contextual de la herramienta de "Rotación 3D"

2. Cursor de datos

Si se pulsa el botón de "Cursor de datos" se sustituye el puntero del ratón por una cruceta que permite crear una etiqueta con coordenadas (X, Y, Z) de la posición seleccionada con tan sólo pulsar con el botón izquierdo del ratón sobre cualquier posición de la flecha simbólica representada.

Si se pulsa el segundo botón sobre los ejes se despliega un menú secundario que permite manejar diversas opciones. Véase figura 113 y 114.

- **Create New Datatip:** crea una nueva etiqueta de coordenadas.
- **Delete Currente Datatip:** elimina la etiqueta de coordenadas seleccionada.
- **Delete All Datatips:** elimina todas las etiquetas de coordenadas.
- **Export Cursor Data to Workspace…:** al seleccionar una etiqueta se pueden exportar sus coordenadas al Workspace.
- **Edit Text Update Function…:** edita una función para actualizar las coordenadas.
- **Select Text Update Function…:** selecciona una función para actualizar las coordenadas.

En el menú secundario "Selection Style", es decir, selección de estilo, se despliega un sub-menú con las siguientes opciones:

- **Mouse Position:** selecciona la posición marcada por el puntero del ratón para indicar las coordenadas espaciales.
- **Snap to Nearest Data Vortex:** selecciona la posición marcada más cercana a un vértice para indicar las coordenadas espaciales.

En el menú secundario "Display Style", es decir, estilo del display, se despliega un sub-menú con las siguientes opciones:

- **Window Inside Figure:** la etiqueta con las coordenadas espaciales se crea dentro de los ejes de representación.
- **Datatip:** la etiqueta con las coordenadas espaciales se crea en una nueva ventana en el exterior de la interfaz gráfica.

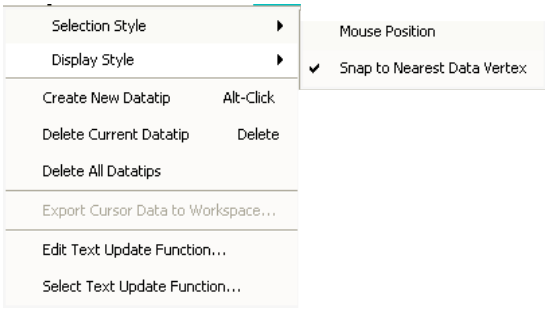


Figura 113. Menú contextual de la herramienta "Cursor de datos" (1)

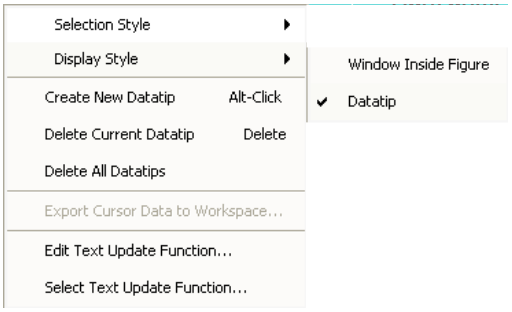


Figura 114. Menú contextual de la herramienta "Cursor de datos" (2)

3. Vista panorámica

Si se pulsa el botón de "Vista panorámica" se sustituye el puntero del ratón por una mano que permite cambiar el punto de vista horizontal y vertical de los ejes con tan sólo mantener pulsado el botón izquierdo del ratón y mover el puntero por cualquier punto del eje cartesiano.

Este botón de la paleta de herramientas sólo se puede utilizar para las representaciones en dos dimensiones.

Si se pulsa el segundo botón del ratón sobre los ejes se despliega un menú secundario que permite manejar algunas opciones interesantes. Véase figura 115.

- **Reset to Original View:** permite regresar a los ejes a su posición inicial de visión establecida.

En el menú secundario “Pan Options”, es decir, opciones de visión panorámica, se despliega un sub-menú con las siguientes opciones:

- **Unconstrained Pan:** permite cambiar simultáneamente el punto de vista horizontal y vertical sin necesidad de barras de desplazamiento.

- **Horizontal Pan:** permite cambiar sólo el punto de vista horizontal.

- **Vertical Pan: permite** cambiar sólo el punto de vista vertical.

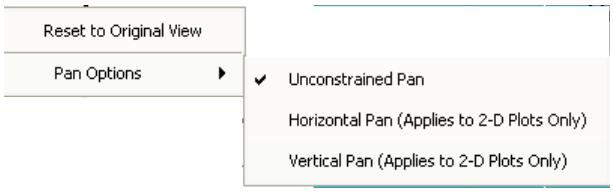


Figura 115. Menú contextual de la herramienta "Vista Panorámica"

Funcionamiento y aplicación de los menús de la interfaz gráfica

En primer lugar, se encuentra el menú “Representación Real” que permite al usuario que maneje en la interfaz la opción de escoger entre una monitorización de la posición real de la máquina tuneladora en dos dimensiones o en tres dimensiones.

Este menú permite al usuario brindar la posibilidad de escoger la representación adecuada en el momento indicado para llevar un control exhaustivo de la trayectoria real que lleva la máquina tuneladora en todo momento. Véase figura 116.

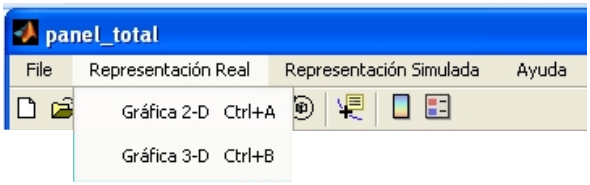


Figura 116. Opciones del menú "Representación Real" de la interfaz gráfica

En segundo lugar, se encuentra el menú "Representación Simulada". La función de este menú es exactamente la misma que el anterior con la diferencia de que el tipo de datos que se pretende monitorizar son los de la simulación generada. Del mismo modo que el menú anterior, permite escoger entra una monitorización de la posición simulada de la máquina tuneladora tanto en dos como en tres dimensiones. Véase figura 117.

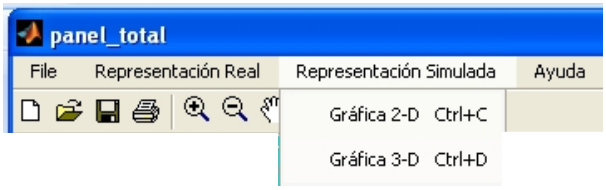


Figura 117. Opciones del menú "Representación Simulada" de la interfaz gráfica

Por último, se encuentra el menú de "Ayuda". Este menú está compuesto por diferentes opciones que permiten al usuario obtener información rápida y práctica del manejo de la interfaz gráfica y de las funciones de botones, tablas y controladores. Véase figura 118.

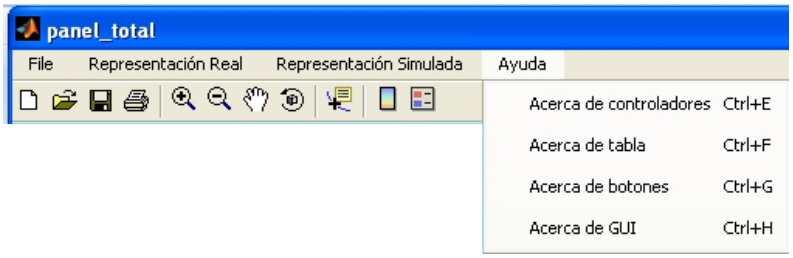


Figura 118. Opciones del menú "Ayuda" de la interfaz gráfica

Funcionamiento y aplicación de la tabla de registros de guiado

La tabla de registros de guiado es fundamental para que el usuario que controla la monitorización de proceso de tunelación pueda realizar un seguimiento paramétrico completo de cada avance que se produce en la máquina tuneladora.

Los valores paramétricos mostrados en la tabla son fundamentales ya que aparte de que éstos contribuyen a una percepción más estricta de la posición en que se encuentra la máquina tuneladora sirven para realizar comparaciones entre los diferentes avances para lograr pronosticar las tendencias y cambios de trayectoria que adopta ésta. La tabla de registros de guiado puede presentar tanto datos reales como datos simulados. Véase figura 119.

La presentación de los datos reales y simulados no se alternan en la misma tabla de registros, sino que para cada tipo de representación, real o simulada, se

cargan los valores en tablas de registro independientes. Es decir, aunque se combinen las representaciones reales y simuladas, los datos reales disponen de una tabla específica para su almacenamiento y los datos simulados disponen de otra tabla independiente.

Registro de guiado

	Vertical	Horizontal	R. Vertical	R. Horizontal	P.K. delantero	P.K. posterior
Avance 1	0	0	0	0	4.7000	0
Avance 2	-105.7460	105.7341	-35.2487	35.2447	7.0484	2.3495
Avance 3	-281.9577	281.8308	-140.9789	140.9154	9.3915	4.6958
Avance 4	-528.5716	528.0365	-317.1429	316.8219	11.7262	7.0357
Avance 5	-845.4925	843.9711	-563.6617	562.6474	14.0493	9.3662
Avance 6	-1.2326e+03	1.2291e+03	-880.4241	877.9490	16.3576	11.6840
Avance 7						

Figura 119. Tabla de registro de guiado

A continuación, se explican cada uno de los valores paramétricos que contiene la tabla de registros de guiado.

- **Vertical:** coordenada "y" de la parte delantera de la máquina.
- **Horizontal:** coordenada "x" de la parte delantera de la máquina.
- **R. Vertical:** coordenada "y" de la parte posterior de la máquina.
- **R. Horizontal:** coordenada "x" de la parte posterior de la máquina.
- **P.K. delantero:** coordenada "z" de la parte delantera de la máquina.
- **P.K. posterior:** coordenada "z" de la parte posterior de la máquina.
- **P.K. central:** coordenada "z" de la parte central de la máquina.
- **Inclinación:** tendencia vertical que sigue la máquina.
- **Ángulo direcc.:** tendencia horizontal que sigue la máquina.
- **Amplitud:** porcentaje que ha variado la intensidad de la señal del haz.
- **Diámetro:** diámetro del haz láser recibido en la tarjeta.
- **Cabeceo:** ángulo de giro de la máquina sobre el eje "x".
- **Guiñada:** ángulo de giro de la máquina sobre el eje "y".
- **Giro:** ángulo de giro de la máquina sobre el eje "z".
- **Avance:** número de avance en que se encuentra la obra.
- **Hora:** hora del actual avance.
- **Fecha:** fecha del actual avance.

4.2 Ejemplo de uso, resultados experimentales

Esta sección está dedicada a abordar diferentes casos prácticos reales o simulados para que el usuario que controle la interfaz gráfica externa sepa interpretar en todo momento tanto los valores paramétricos como la monitorización de la trayectoria de la máquina tuneladora para poder actuar con previsión y anticipación a la hora de corregir dicha trayectoria adoptada.

En primer lugar, cada vez que se inicializa la aplicación de la interfaz gráfica, se configuran las condiciones iniciales y los sistemas de referencia para que el primer avance monitorizado simulado en dos dimensiones por la interfaz gráfica, posicione a la flecha simbólica que tipifica la máquina en el centro geométrico de los ejes. No se producen ni giros angulares, ni desplazamientos de la flecha. Véase figura 120.

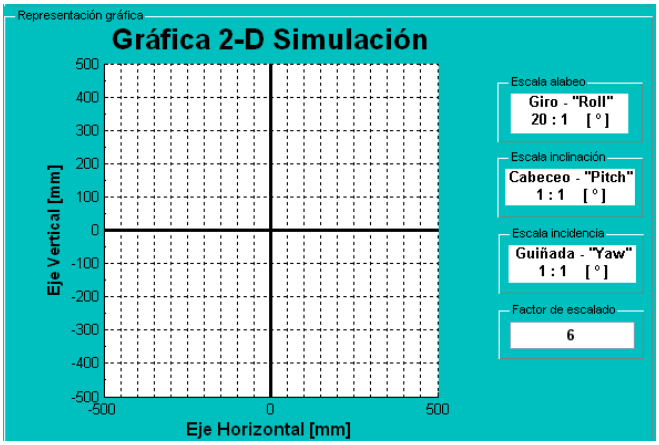


Figura 120. Representación gráfica de la trayectoria rectilínea de la máquina tuneladora en dos dimensiones

En la figura 121, se visualiza la monitorización simulada en tres dimensiones de la máquina tuneladora adoptando una trayectoria rectilínea central.

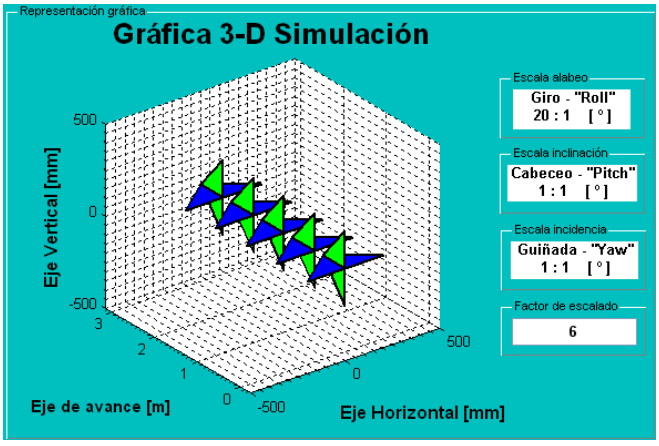


Figura 121. Representación gráfica de una trayectoria rectilínea de la máquina tuneladora en tres dimensiones

Los siguientes avances que se produzcan en el proceso de tunelación pueden adoptar múltiples trayectorias.

A continuación, se van a explicar detalladamente todas las direcciones posibles que puede asumir la máquina tuneladora durante la excavación.

En primer lugar, la máquina tuneladora puede asumir un desplazamiento y giro vertical ascendente o descendente. Es decir, la parte delantera de la máquina tuneladora se eleva y desciende de la parte posterior.

Esto provoca que el ángulo de giro de la máquina sobre el eje "X" disminuya o aumente su valor produciéndose una rotación sobre dicho eje. Este fenómeno es conocido como inclinación o cabeceo de la máquina tuneladora.

El valor "Pitch" tipifica la magnitud del giro sobre el eje "X" en grados y se puede apreciar en la tabla de registros de guiado.

Los valores "Vertical" y "Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte delantera de la máquina y los valores "R. Vertical" y "R. Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte posterior de la máquina tuneladora. Estos valores se pueden apreciar tanto en los paneles paramétricos como en la tabla de registros de guiado.

En la figura 122, se visualiza la monitorización simulada en dos dimensiones de la máquina tuneladora adoptando un giro sobre su eje transversal.

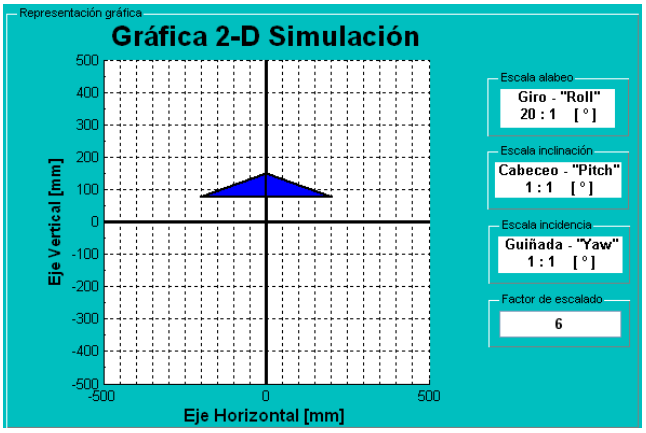


Figura 122. Representación gráfica de una trayectoria de elevación en dos dimensiones

En la figura 123, se visualiza la monitorización simulada en tres dimensiones de la máquina tuneladora adoptando una trayectoria de giro sobre su eje trasversal.

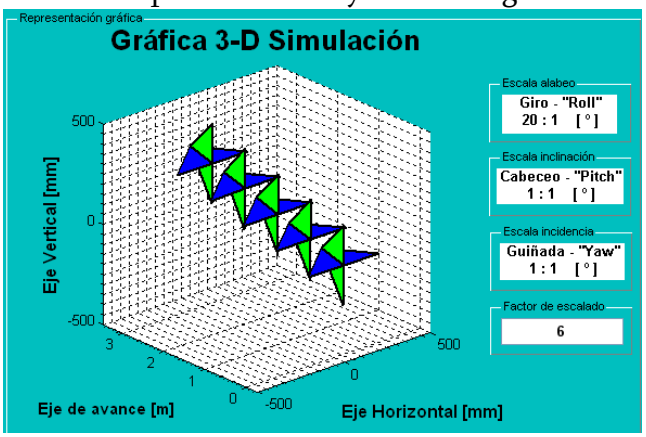


Figura 123. Representación gráfica de una trayectoria de elevación en tres dimensiones

En segundo lugar, la máquina tuneladora puede asumir un desplazamiento y giro horizontal positivo o negativo. Es decir, la parte delantera de la máquina tuneladora se posiciona a la derecha o izquierda de la parte posterior. Esto provoca que el ángulo de giro de la máquina sobre el eje "Y" disminuye o aumenta su valor produciéndose una rotación sobre dicho eje. Este fenómeno es conocido como incidencia o guiñada de la máquina tuneladora.

El valor "Yaw" tipifica la magnitud del giro sobre el eje "Y" en grados y se puede apreciar en la tabla de registros de guiado.

Los valores "Vertical" y "Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte delantera de la máquina y los valores "R. Vertical" y "R. Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte posterior de la máquina tuneladora. Estos valores se pueden apreciar tanto en los paneles paramétricos como en la tabla de registros de guiado.

En la figura 124, se visualiza la monitorización simulada en dos dimensiones de la máquina tuneladora adoptando un giro sobre su eje vertical.

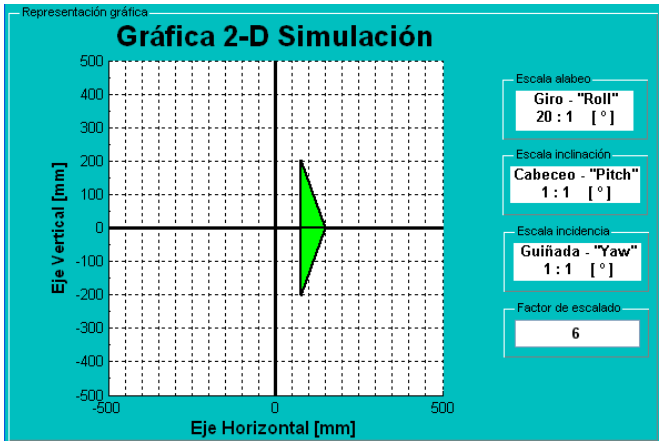


Figura 124. Representación gráfica de una trayectoria de giro horizontal en dos dimensiones

En la figura 125, se visualiza la monitorización simulada en tres dimensiones de la máquina tuneladora adoptando una trayectoria de giro sobre su eje vertical.

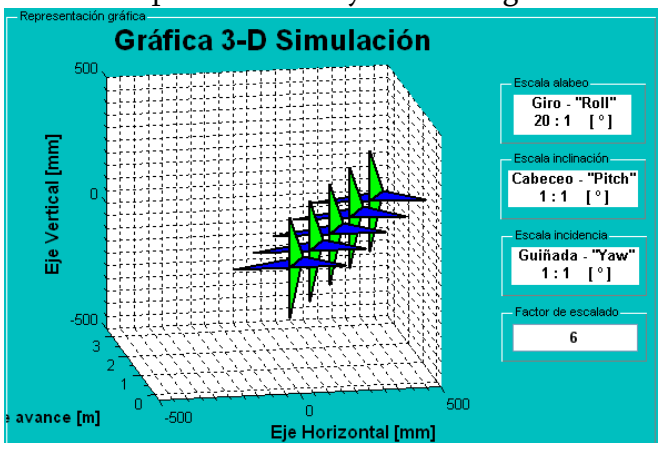


Figura 125. Representación gráfica de una trayectoria de giro horizontal en tres dimensiones

En tercer lugar, la máquina tuneladora puede asumir una trayectoria diagonal, es decir, se alternan simultáneamente el desplazamiento y giro vertical y horizontal. Se pueden generar cuatro direcciones diferentes en función del ángulo de giro que adopte la máquina tuneladora sobre los ejes "X" e "Y". Este fenómeno es conocido como inclinación más incidencia.

Los valores "Pitch" y "Yaw" se puede apreciar en la tabla de registros de guiado y se expresan en grados.

Los valores "Vertical" y "Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte delantera de la máquina y los valores "R. Vertical" y "R. Horizontal" tipifican el desplazamiento de las coordenadas "X" e "Y" de la parte posterior de la máquina tuneladora. Estos valores se pueden apreciar tanto en los paneles paramétricos como en la tabla de registros de guiado.

En la figura 126, se visualiza la monitorización simulada en dos dimensiones de la máquina tuneladora adoptando un giro sobre sus ejes trasversal y vertical.

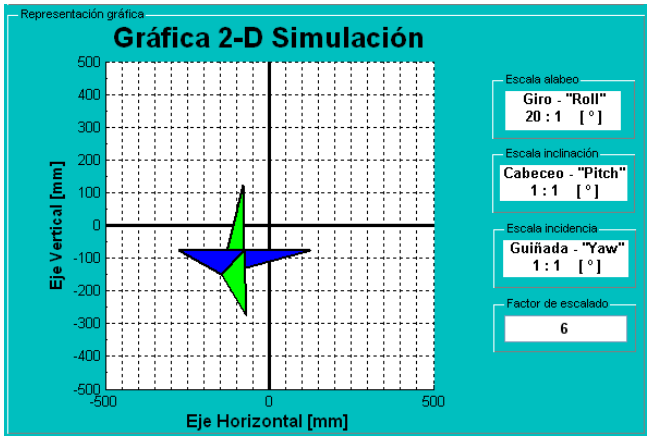


Figura 126. Representación gráfica de una trayectoria de giro horizontal y vertical en dos dimensiones

En la figura 127, se visualiza la monitorización simulada en tres dimensiones de la máquina adoptando una trayectoria de giro sobre sus ejes trasversal y vertical.

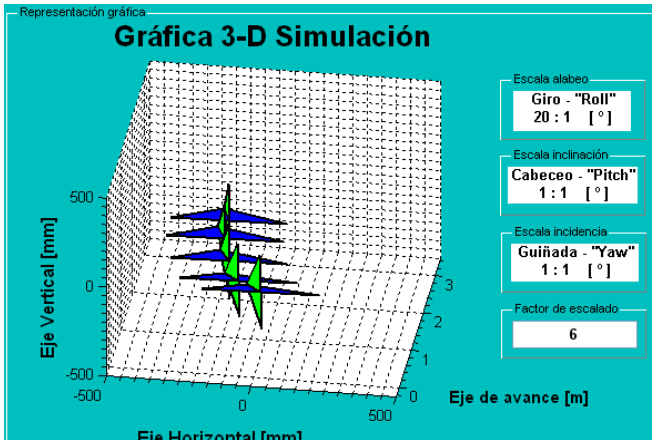


Figura 127. Representación gráfica de una trayectoria de giro horizontal y vertical en tres dimensiones

En cuarto lugar, la máquina tuneladora puede asumir un movimiento rotativo sobre su propio eje, es decir, una desviación angular sobre sí misma. Esta situación se ocasiona debido a que la máquina tuneladora se ha quedado obstruida por la dureza del terreno y gira sobre sí misma ya que le resulta imposible avanzar. En cuanto el usuario detecte en los paneles paramétricos que se produce un giro de la máquina superior o inferior a tres grados, debe detener el proceso de tunelización de inmediato ya que se producirían grandes daños en la cabeza cortadora de la máquina tuneladora.

Este fenómeno es conocido como alabeo o "Roll" y se puede apreciar tanto en la tabla de registros de guiado como en los paneles paramétricos debido a su importancia.

En la figura 128, se visualiza la monitorización simulada en dos dimensiones de la máquina tuneladora adoptando un giro sobre su eje longitudinal.

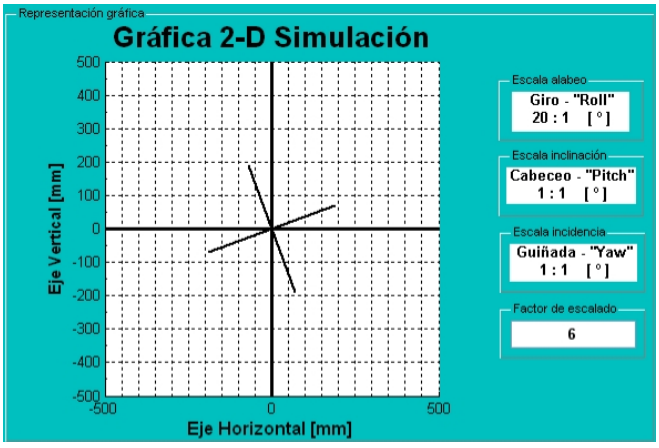


Figura 128. Representación gráfica de una trayectoria de giro sobre el eje longitudinal en dos dimensiones

En la figura 129, se visualiza la monitorización simulada en tres dimensiones de la máquina adoptando una trayectoria de giro sobre su eje longitudinal.

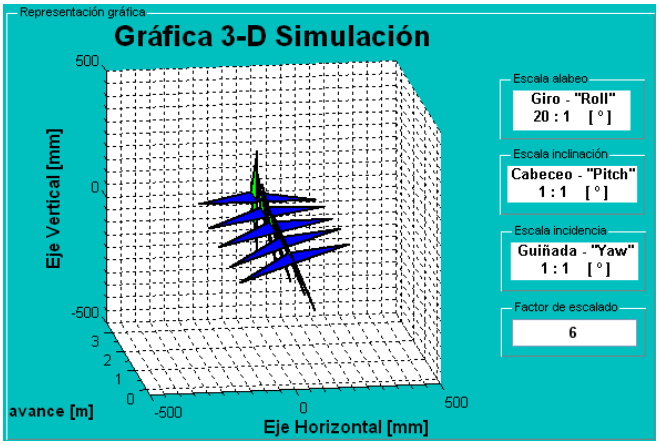


Figura 129. Representación gráfica de una trayectoria de giro sobre el eje longitudinal en tres dimensiones

Por último, la máquina tuneladora puede asumir un desplazamiento posicional, es decir, que realice una trayectoria rectilínea desde una posición diferente a la inicial sin que se produzcan desviaciones angulares. Esta situación se ocasiona debido a que la máquina tuneladora puede tunelar unos metros para ganar o perder profundidad y acto seguido volver a la trayectoria rectilínea inicial para realizar el túnel correctamente.

En la figura 130, se visualiza la monitorización simulada en dos dimensiones de la máquina tuneladora adoptando sólo un movimiento de desplazamiento vertical y horizontal.

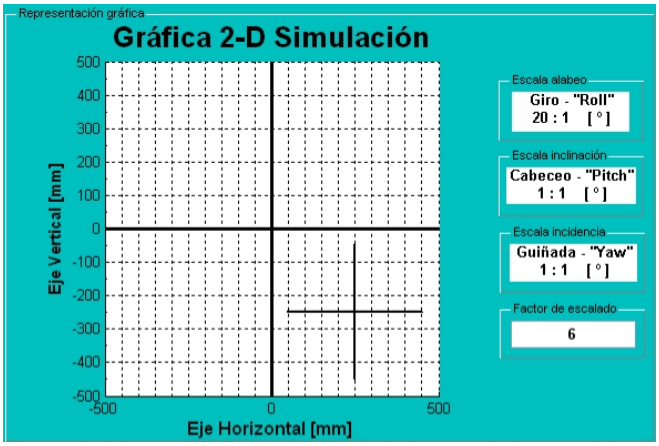


Figura 130. Representación gráfica de una trayectoria desplazada en dos dimensiones

En la figura 131, se visualiza la monitorización simulada en tres dimensiones de la máquina tuneladora adoptando una trayectoria rectilínea sólo de un movimiento de desplazamiento vertical y horizontal.

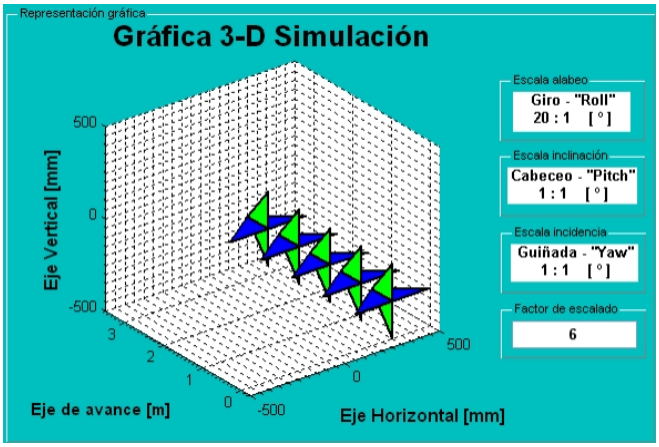


Figura 131. Representación gráfica de una trayectoria desplazada en tres dimensiones

Una vez finalizado el proceso de tunelización, se pulsa el botón de "Salir" en los paneles de control de la interfaz gráfica para abandonar la aplicación. Acto seguido, los datos almacenados en la tabla de registro de guiado por parte de las representaciones reales y simuladas son exportados a dos ficheros Excel. Estos ficheros Excel se encuentran almacenados en dos carpetas situadas en el directorio de trabajo del software de MATLAB. Estas carpetas se denominan "Registros_reales" y "Registros_simulados". Véase figura 132.

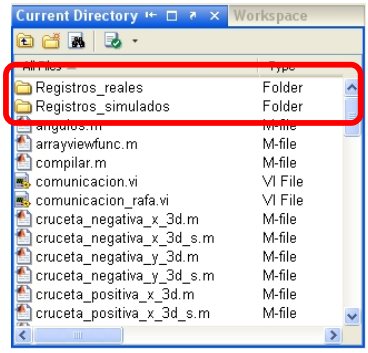


Figura 132. Carpetas donde se almacenan los registros de guiado reales y simulados

Los ficheros Excel generados quedan nombrados con la fecha y hora del momento exacto en que se finaliza la aplicación y además van precedidos del pseudónimo de "Real" o "Simulado" que adoptan dependiendo de la representación elegida. Véase figura 133 y 134.

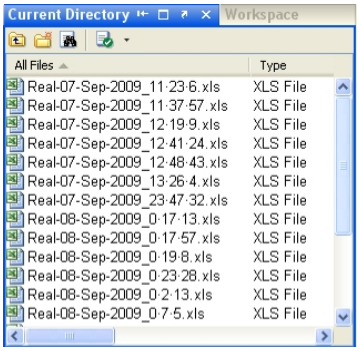


*Figura 133. Registros de guiado reales *.xls*

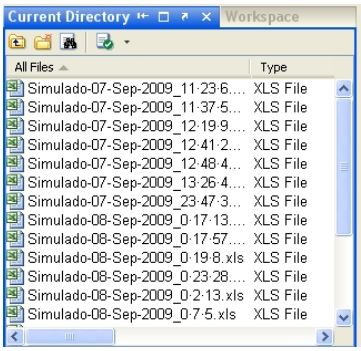


*Figura 134. Registros de guiado simulados *.xls*

Para finalizar, los archivos Excel se generan automáticamente, independientemente de que se hayan realizado representaciones o no. Es decir, si se está realizando una simple prueba con el simulador y no se utiliza el menú de representaciones reales, el software generará ambos ficheros Excel con la peculiaridad de que el fichero de datos reales se encontrará vacío de valores paramétricos. Esto se configura así para poder acceder a los registros de guiado en cualquier momento y conocer todos los detalles de ambas representaciones.

4.3 Consejos y advertencias

- La monitorización de las representaciones gráficas en dos y tres dimensiones no son objetivas por norma general con respecto a las dimensiones reales de la máquina tuneladora. Las flechas simbólicas representan un pequeño fragmento de la máquina real, es decir, tipifican una reducción mayor o menor de la máquina tuneladora dependiendo del factor de escalado que se introduzca. El factor de escalado puede ser variado manualmente por el operario para la mejor visualización gráfica. Para permitir representar en una misma gráfica cinco flechas simbólicas yuxtapuestas sin que se sobrepongan entre ellas, se deja una distancia entre las flechas de un tercio de su longitud total escalada. De esta manera, permiten al usuario dar una imagen visual de la trayectoria que adopta la máquina tuneladora.

- El factor de escalado se realiza debido a que la máquina tuneladora real suele seguir una trayectoria bastante rectilínea donde se producen pocas desviaciones y cambios de trayectoria. Por ese motivo, el escalamiento reductor ayuda a amplificar esas desviaciones y cambios de trayectoria para que el usuario que controla la interfaz gráfica, visualice más nítidamente dichas desviaciones y pueda aplicar con antelación las correcciones oportunas en el guiado de la máquina tuneladora.

- Del mismo modo, se realiza un escalado del ángulo de giro del eje longitudinal de la máquina tuneladora "Roll", para amplificar visualmente las desviaciones angulares que se producen, con un factor de ampliación variable de la condición inicial establecida.

- Los valores tanto de los paneles paramétricos como de la tabla de registro de guiado son los valores reales de las dimensiones totales de la máquina tuneladora. Estos valores son reales para que el usuario que controle la interfaz gráfica externa sea consciente de las dimensiones reales de la máquina y sólo utilice las representaciones gráficas como ayuda en la corrección de la dirección o trayectoria que adopta la máquina tuneladora. Por tanto, los valores expuestos tanto en los paneles paramétricos como en la tabla de registro de guiado no coinciden con las representaciones gráficas monitorizadas.

- No todas las obras y túneles son iguales. Hay que tener en cuenta que en cada excavación se trabaja con distintos tipos de terreno y por lo tanto el guiado de una máquina tuneladora varía ostensiblemente. Es importante destacar que la máquina tuneladora tiene tendencia a elevarse ya que avanza por donde tiene más facilidad y por donde el terreno sea más blando.

- Es imprescindible tener una referencia previa por parte de los ingenieros topógrafos de la trayectoria ideal que pretende seguir la máquina tuneladora para poder anticipar maniobras y mantener el guiado lo más cercano posible a la trayectoria ideal.

- Es importante llevar un seguimiento tanto de los paneles paramétricos como de la tabla de registros de guiado para poder prever desviaciones de la trayectoria prevista. Además es necesario tener anticipación a la hora de realizar cambios de trayectoria ya que la máquina tuneladora necesita avanzar mucho terreno ya que posee un radio de curvatura muy bajo.

- El ángulo de rotación de la máquina sobre el eje longitudinal o alabeo ("Roll"), no debe sobrepasar nunca los ± 3° de ángulo de giro ya que produciría serias averías en la máquina tuneladora. Si por alguna circunstancia sobrepasa los ± 3°, se activa un mensaje de error en la interfaz gráfica para que se detenga por completo la máquina hasta que el personal de la obra consiga reorientar la máquina hasta su posición estipulada.

5. Conclusiones

5.1 Conclusiones

En este apartado, se hará una recapitulación de todos los objetivos propuestos al principio del presente Proyecto de Fin de Carrera y se contrastarán los objetivos iniciales con los resultados finales obtenidos. Además se comentarán las mejoras e innovaciones que han surgido durante el desarrollo de estos objetivos.

Se ha conseguido llevar a cabo el objetivo principal de desarrollar una interfaz gráfica externa, réplica de los sistemas actuales, que sirva para la monitorización del guiado de las máquinas tuneladoras.

Se han realizado mejoras tanto en la visualización de las representaciones gráficas de la posición espacial de la máquina tuneladora como en la monitorización de los parámetros calculados gracias a la tabla de registros de guiado. Esta tabla de registros almacena los valores más significativos que el operario necesita. Es imprescindible para poder realizar un guiado predictivo de la máquina tuneladora y poder anticipar maniobras de corrección de trayectoria para mejorar el guiado.

Además, se han añadido menús interactivos que permiten al usuario seleccionar el modo de representación gráfica que desea visualizar, dos o tres dimensiones, y el tipo de datos entrantes con el que se desea trabajar, reales o simulados. También, se ha creado un menú de ayuda para obtener en cualquier momento la información necesaria con todo lo que respecta a los elementos de la interfaz gráfica.

Se ha ampliado la funcionalidad de la interfaz gracias a la inserción de paletas de herramientas gráficas y la creación de botones funcionales para que el usuario pueda interactuar.

Se ha conseguido realizar un escalado variable de la máquina tuneladora tipificándola en forma de flecha simbólica para representarla como referencia tanto en dos dimensiones como en tres dimensiones. Gracias a éste escalado manual de las dimensiones totales de la máquina, se ha conseguido ampliar o reducir la visión de las dimensiones de ésta, para apreciar con mayor precisión los cambios de dirección y trayectoria que se producen en el guiado de la máquina.

Del mismo modo, se ha realizado un escalado de ampliación en el ángulo de giro sobre el eje longitudinal para aumentar las desviaciones angulares sobre dicho eje y que sean apreciables en la interfaz gráfica.

Se ha realizado el almacenaje de todos los valores paramétricos en un archivo Excel después de finalizar la aplicación de la interfaz gráfica, para poder comparar el último registro guardado, con la previsión de trayectoria ideal prevista para el proceso de tunelación.

Todo ello, ha contribuido para que la interfaz gráfica se convierta en una aplicación mucho más interactiva y permita al usuario un gran abanico de posibilidades para optimizar el sistema de guiado y mejorar el proceso de monitorización de las máquinas tuneladora.

La interfaz gráfica comunicada con el simulador de hinca de tubos se ha convertido en la perfecta herramienta o consola virtual para la formación de futuros controladores de máquinas tuneladoras. Tiene un nivel de realismo muy alto ya que se dispone de la misma interfaz gráfica tanto para realizar una monitorización real como para monitorizar una situación simulada.

Por tanto, la interfaz gráfica de guiado se ha convertido en un sistema fundamentalmente visual e intuitivo, para que el operario que esté controlando la aplicación, pueda tener en todo momento un conocimiento rápido y preciso de la posición y situación actual en que se encuentra la máquina tuneladora.

Las comunicaciones entre los diferentes módulos implicados en el sistema de guiado han sido correctamente sincronizadas para realizar automáticamente todo el flujo de intercambios de variables sin producirse ningún fallo de lectura o escritura de datos.

El tiempo de ejecución del software se ha depurado totalmente para que las variables almacenadas en los ficheros de intercambio de datos lleguen rápidamente a la interfaz gráfica externa para que ésta pueda monitorizar los avances producidos por la máquina tuneladora prácticamente en tiempo real.

5.2 Análisis crítico

En este apartado se hará un análisis crítico de los problemas que han surgido durante el desarrollo del proyecto y se analizará cuáles de éstos se han podido subsanar y cuáles de ellos quedarán como trabajos futuros y ampliaciones.

En primer lugar, surgió el problema de intentar implementar en una misma interfaz gráfica los dos tipos de representación, dos dimensiones y tres dimensiones. En principio, parecía una opción inviable debido a que la programación del software de MATLAB seguía una programación mono-hilo y no podía llevar a cabo las dos representaciones en procesos diferentes. Además debían estar sincronizados tanto los modos de representación gráfica como la alternancia de variables reales o simuladas. Estos cuatro factores que había que tener en cuenta creaban un abanico de posibilidades demasiado grande para poder controlar todas las situaciones a la hora de intercambiar de un modo a otro, y no producir errores y pérdidas de información.

Este problema de solucionó utilizando el recurso de las celdas de almacenamiento. Estas celdas permiten almacenar tantas filas de variables como avances tenga el proceso de tunelización de la trayectoria proyectada. Por tanto, se creó una celda de almacenamiento para cada modo de representación gráfica y para cada tipo de dato entrante (real o simulado). De esta manera, se pudo mantener un control y almacenamiento en cada avance de todas las variables entrantes que permitió representar en cada momento lo que el usuario solicitaba en la interfaz gráfica externa.

En segundo lugar, surgió el problema de la inserción en la interfaz gráfica de una tabla de registros de guiado. Esta tabla debía registrar todos los avances que le llegaran y mantener en su registro todas las variables de cada uno de los avances para dar al usuario la opción de realizar comparaciones con los anteriores avances almacenados. En consecuencia, surgió el primer problema ya que no se quedaban almacenados los datos en la tabla entre un avance y otro.

Además, debía mantenerse una independencia en la tabla entre los datos reales y los datos simulados, es decir, tendría que haber una tabla para cada tipo de datos. Por tanto, surgió el segundo problema, tener dos tipos de datos y una sola tabla de registros.

Ambos problemas se solucionaron de nuevo con la creación de celdas de almacenamiento. Se creó una celda de almacenamiento para datos reales y otra para datos simulados. Estas celdas tipifican una tabla de registros para cada tipo de dato y tienen almacenados todos los parámetros de los avances de forma independiente. Por tanto, sólo quedaba monitorizar cada una ellas dependiendo del modo de visualización elegido por el usuario.

En tercer lugar, surgió el problema de la comunicación por librería dinámica con el resto de los módulos del sistema de guiado. Se comenzó analizando todos los métodos de comunicación posibles y cuál era el más ventajoso. Se decidió que la librería dinámica era el método más efectivo por rendimiento y eficiencia. Sin embargo, los problemas surgieron desde el primer momento, ya que no se disponía apenas de información útil y los documentos de ayuda disponibles no había manera posible de llevarlos a la práctica.

Por tanto, después de tres meses de investigación con la ayuda del personal docente del Departamento de Ingeniería de Sistemas y Automática, se decidió afrontar otro método de comunicación de menor rendimiento pero más efectivo, resultando ser más práctico, intuitivo y con la misma funcionalidad y eficiencia.

En cuarto lugar, surgió el problema de la visualización de la tabla de registros al crear una aplicación ejecutable externa. La creación de un ejecutable del software de la GUI creada en MATLAB no debía ser en principio un problema ya que el propio programa viene implementado para realizar este tipo de operaciones. Sin embargo, después de generar la aplicación ejecutable se descubrió que la tabla de registros de guiado no representaba los datos almacenados en las celdas de almacenamiento.

Por tanto, para solucionar el problema se añadieron al directorio de trabajo de la GUI dos funciones definidas de MATLAB necesarias para realizar la compilación. Los errores que se generaban al realizar la compilación remitieron pero al ejecutarse la aplicación externa los problemas seguían acentuándose al no poder visualizarse los valores en la tabla de registros.

Este problema no se ha conseguido remediar de momento, pero la solución no tardará mucho en llegar cuando se produzca una actualización de funciones del compilador del programa MATLAB 2008a.

Cabe destacar, que el balance total del análisis crítico sale positivo, ya que prácticamente todos los problemas que han frenado el desarrollo del proyecto, se han ido desatascando invirtiendo muchas horas de investigación y muchas horas de sueño.

5.3 Trabajos futuros y ampliaciones

Los resultados obtenidos han cumplido con rigor con la expectativas marcadas por los objetivos iniciales del Proyecto de Fin de Carrera pero lo cierto es que es difícil poner fin a un proyecto de esta categoría, pues siempre hay nuevas mejoras que introducir, que conviertan a la interfaz gráfica en una aplicación un poco más real y práctica.

A continuación, se exponen algunas mejoras que podrían ser tenidas en cuenta para una futura optimización de este proyecto:

- Programar un sistema de alertas en el software de la interfaz gráfica que avise al operario que conduzca la máquina tuneladora para evitar posibles situaciones de peligro. Por ejemplo, evitar una desviación excesiva de la trayectoria ideal a seguir, atasco de la máquina tuneladora, fallo en el software del sistema de guiado.

- Mejora de las comunicaciones entre los diferentes módulos del sistema de guiado. Para ello, se podría recurrir a otro método de comunicación detallado anteriormente en el apartado de "Comunicación entre los diferentes módulos software" como librerías de enlace dinámico (DLL) ó comunicación por Socket, incluso recurrir a otro método alternativo.

- Transformar el software actual basado en una arquitectura mono-hilo, en una aplicación multi-hilo, lo cual produciría una significativa mejora en el rendimiento de toda la aplicación, ya que en el modelo actual los procesos se ejecutan secuencialmente con toda la latencia y el retardo que esto conlleva, y con una nueva versión multi-hilo todos los procesos se ejecutarían concurrentemente, lo cual dotaría al sistema de mucha más fluidez a la hora de trabajar con él.

- Trasladar todo el software generado con MATLAB a otros lenguajes gráficos más potentes y con muchas más posibilidades, por ejemplo, "Delphi" o "Visual Basic".

- Mejorar la visualización y la facilidad de manejo del sistema de guiado creando un sistema de realidad virtual sobre la base de la interfaz gráfica que permita introducir una trayectoria ideal del recorrido a tunelar para que el operario que conduzca la máquina tuneladora pueda realizar constantes comparaciones visuales para no desviarse en exceso de la trayectoria marcada y sin tener que prestar atención a la información paramétrica de la tabla de registro de guaido.

- Diseñar un sistema multi-pantalla donde se pueda monitorizar la progresiva evolución de los avances de la máquina tuneladora desde el punto de vista de dos y tres dimensiones simultáneamente.

- Diseñar una aplicación gráfica en VRML para dar mayor realismo a las representaciones gráficas y al entorno de la tunelación.

- Diseñar una aplicación que permita instruir a los alumnos en el manejo del simulador virtual. Esta aplicación podría incorporar, por ejemplo, la posibilidad de creación de perfiles para los diferentes usuarios, seleccionar diferentes niveles de dificultad, registrar cada una de las sesiones de los usuarios, herramientas de autoevaluación y corrección de los fallos detectados, posibilidad de visualizar las sesiones guardadas, etc. . .

En definitiva, estas propuestas buscan un simulador más veraz, tanto en las posibilidades que ofrezca, como en el modo en que éstas se presenten al usuario, con la intención de proporcionar una herramienta de trabajo y aprendizaje más sólida, que facilite la manipulación del sistema de guiado y reduzca significativamente los plazos y costes de formación de pilotos para estas máquinas.

6. Bibliografía

MANUALES

[1] GARCÍA DE JALÓN, Javier; IGNACIO RODRÍGUEZ, José; VIDAL, Jesús. "Aprenda Matlab 7.0 como si estuviera en primero". Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid. Diciembre de 2005.
http://mat21.etsii.upm.es/ayudainf/aprendainf/Matlab70/matlab70primero.pdf

[2] CASADO FERNÁNDEZ, MªCristina. "Manual Básico de Matlab". Servicios Informáticos U.C.M. Apoyo a Investigación y Docencia. Universidad Complutense de Madrid.
http://www.sisoft.ucm.es/Manuales/MATLAB_r2006b.pdf

[3] BARRAGÁN GUERRERO, Diego Orlando. "Manual de interfaz gráfica de usuario en Matlab. Parte I". Ecuador.
http://www.matpic.com/MATLAB/MATLAB_GUIDE.html

[4] HAUGEN, Finn. "Introduction to LabVIEW 8.5". Agosto de 2008
http://techteach.no/labview/lv85/labview/index.htm

[5] HOW TO GENERATE A DLL FROM A MATLAB FUNCTION SCRIPT TO RUN IN LABVIEW
http://forums.ni.com/ni/attachments/ni/170/29800/1/dllmatlablabview.pdf

PÁGINAS WEB

[6] VMT GmbH – Gesellschaft für Vermessungstechnik. "Sistema de guiado para hinca de tubo". Diciembre de 2003.
http://www.microtunel.com/descargas/guiado_hinca_slsrv.pdf

[7] MARTÍ CARDONA, MARC. "Microtunel". Actualización 01/06/2008
http://www.microtunel.com/41_vmt_es.htm

[8] VMT GmbH. "Sistema de guiado - SLS-Microtunnelling LT"
http://www.vmt-gmbh.de/271.html?&L=1

SOFTWARE

[9] CORKE , Peter I. "Robotics Toolbox for Matlab". Abril de 2001. http://www.bridgeport.edu/sed/fcourses/cs460_cpe460/Lectures_Handouts/1/Robotics_Toolbox_for_MATLAB_rel6.pdf

TESIS

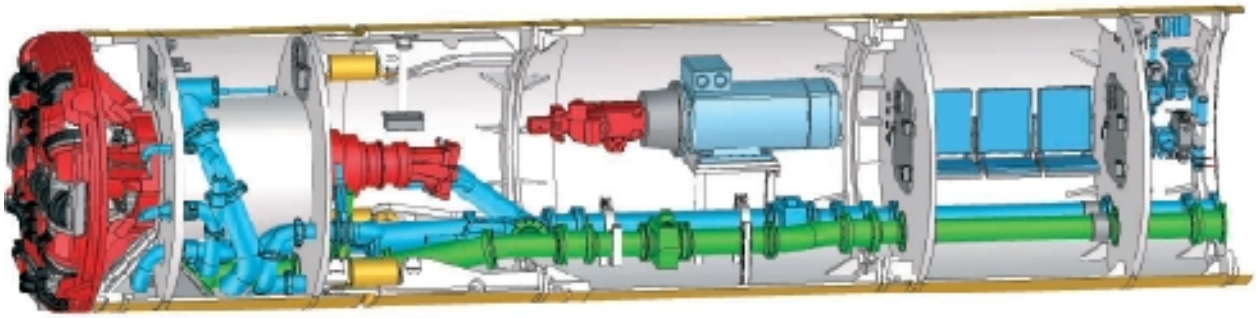
[10] GONZÁLEZ DE LA VEGA, Carlos. “Desarrollo de un simulador básico de hinca de tuberías”- Tutor: Alberto Jardón Huete. Universidad Carlos III de Madrid, Departamento de Ingeniería de Sistemas y Automática. 2009

7. Anexos

ANEXO A. Hoja de características de la máquina tuneladora AVND

AVND 1600 AB – AVND 3000 AB

Pipe Jacking



Special Features

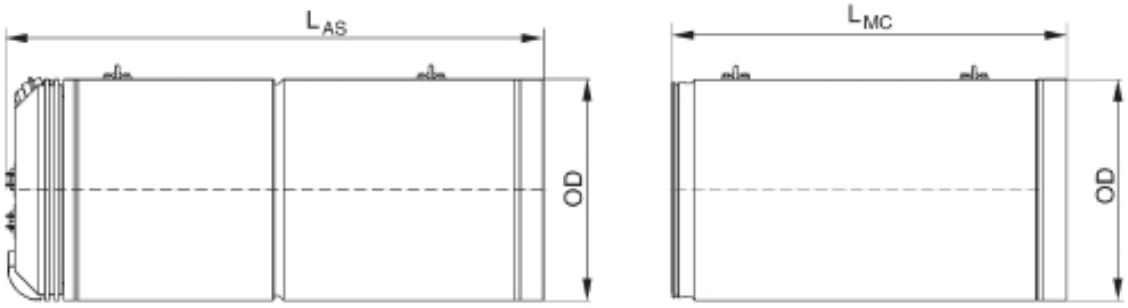
- Hydroshield-mode for inhomogeneous soil conditions or low coverage.
- Alternatively, conventional Slurry-mode.
- Designed for soft ground, mixed ground and hard rock conditions by use of different cutting wheels (full face excavation).
- Access to cutting wheel for tool replacement (hyperbaric system).
- Highly reliable steering operation due to inductive measuring system.
- Different (variable) flushing modes/ jet systems to suit different ground conditions.
- High pressure water system for operation in cohesive soil.
- Providing highly effective cone crusher.
- Equipped with heavy duty long-life main bearing and high torque control drive.
- Hydraulic power pack in machine can
- Completely remote-controlled.
- All machines compatible to VMT Guidance System.

Technical Data		AVND1800AB		AVND1800AB		AVND2000AB		AVND2200AB		AVND2400AB		AVND2800AB		AVND3000AB	
		Std*	Ext*	Std	Ext	Std	Ext	Std	Ext	Std	Ext	Std	Ext	Std	Ext
1. Articulated Shield															
Outer diameter	mm	1970	2 150	2 150	2 425	2 425	3 025	2 725	3 025	3 025	3 625	3 125	3 525	3 625	4 225
Pipe OD	mm	1940	2 120	2 120	2 400	2 400	3 000	2 700	3 000	3 000	3 600	3 100	3 500	3 600	4 200
Pipe ID	mm	1 000	1 000	1 800	2 000	2 000	2 400	2 200	2 500	2 400	3 000	2 000	3 300	3 000	3 000
Main drive															
Max. torque	kNm	360		500		730		780		1 100		1 300		1 600	
Revolution LH / RH	rpm	0-4.6		0-4.6		0-8.4		0-8.4		0-8.8		0-8.8		0-8.1	
Rated Power	kW	110		160		315		315		315		315		400	
Roll correction		✓		✓		✓		✓		✓		✓		✓	
Steering															
Steering cylinders		4		4		4		4		8		8		8	
Force per cyl. / oil pressure	kN/bar	1 606/500		1 606/500		1 272/500		1 272/500		770/500		770/500		1 005/500	
Stroke per cyl.	mm	100		100		100		100		150		150		150	
Control															
Computer data logging system		✓		✓		✓		✓		✓		✓		✓	
Fuzzy control (automatic steering)		opt.		opt.		opt.		opt.		opt.		opt.		opt.	
Fully visualized process control		✓		✓		✓		✓		✓		✓		✓	
Active roll protection (el-hydr)		✓		✓		✓		✓		✓		✓		✓	
VMT System:	GLS	✓		✓		✓		✓		✓		✓		✓	
(Suitability)	GLS-HLW	✓		✓		✓		✓		✓		✓		✓	
	INS-F	✓		✓		✓		✓		✓		✓		✓	
	GNS P	✓		✓		✓		✓		✓		✓		✓	
2. Machine Can															
Lubrication System		✓		✓		✓		✓		✓		✓		✓	
3. General Information															
Pipe jacking		✓		✓		✓		✓		✓		✓		✓	
Drive length (recommended)	m	700		900		1 100		1 100		1 100		1 100		1100	
Access to cutting wheel		✓		✓		✓		✓		✓		✓		✓	
Watertightness	bar	3		3		3		3		3		3		3	
Airlock		✓		✓		✓		✓		✓		✓		✓	
Telescopic and interjacking station		opt.		opt.		opt.		opt.		opt.		opt.		opt.	
Slurry line diam.	mm	150		150		200		200		200		200		250	
High pressure water system		✓		✓		✓		✓		✓		✓		✓	
Telescopic and interjacking station		opt.		opt.		opt.		opt.		opt.		opt.		opt.	

All measures and data represent the main feasibility of the machines. Individual solutions are possible. Errors excepted.

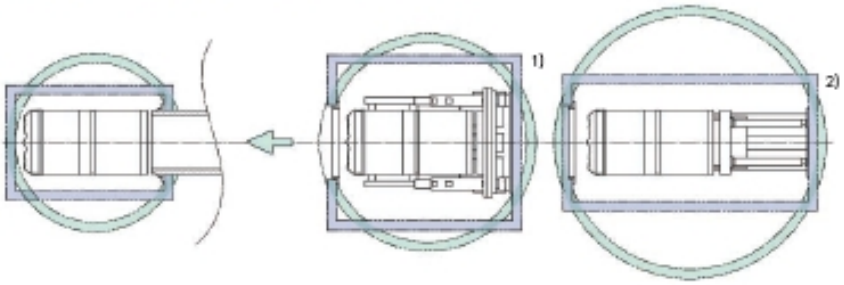
* Std – standard; Ext – extension kit

Machine dimensions



			AVND1600AB	AVND1800AB	AVND2000AB	AVND2200AB	AVND2400AB	AVND2600AB	AVND3000AB
Outer Diameter	OD	mm	1 970	2 150	2 425	2 725	3 025	3 125	3 625
Length artic. shield	l_{AS}	mm	4 200	4 400	4 700	4 800	4 800	6 900	6 900
Length machine can	l_{MC}	mm	3 200	3 200	3 200	3 200	3 200	–	–
Max. single weight	w	kg	24 000	27 000	38 000	42 000	45 000	60 000	70 000

Shaft dimensions



1) Compact jacking rig:
- smaller launch shaft needed

2) Main jacking station:
- continuous push with telescopic cylinders (time advantage)
- higher thrust capacity possible by adding of main jacks

		AVND1600AB	AVND1800AB	AVND2000AB	AVND2200AB	AVND2400AB	AVND2600AB	AVND3000AB
Launch Shaft	Pipe							
Compact jacking rig	length	Shaft size	Shaft size	Shaft size	Shaft size	Shaft size	Shaft size	Shaft size
	3 000mm	Ø = 7.0m	Ø = 7.5m	Ø = 7.5m	–	–	–	–
		6.5m x 4.5m	7.0m x 4.5m	7.0m x 5.0m	–	–	–	–
Launch Shaft	3 000mm	Ø = 9.0m	Ø = 9.5m	Ø = 9.5m	Ø = 10.0m	Ø = 10.0m	Ø = 12.0m	Ø = 12.0m
Main jacking station		8.5m x 4.0 m	9.0m x 4.0m	9.5m x 4.5m	9.5m x 5.0 m	9.5m x 5.0m	11.5m x 5.5m	11.5m x 6.0m
Reception Shaft	l_{AS}	4 200mm	4 400mm	4 700mm	4 800mm	4 800mm	6 900mm	6 900mm
Circular		Ø = 5.5m	Ø = 5.5m	Ø = 6.0m	Ø = 6.5m	Ø = 6.5m	Ø = 8.5m	Ø = 8.5m
Rectangular		5.0m x 3.0m	5.0m x 3.0m	5.5m x 3.5m	5.5m x 4.0m	6.0m x 4.0m	8.0m x 4.5m	8.0m x 5.0m

All dimensions according to 15m shaft depth.

Machine type description e.g. AVN[1] 1800[2] T[3] B[4]

[1] Machine type	[2] ID of jacking pipe	[3] Access to cutting wheel	[4] Type of container, power transfer from container to machine
		X = no access	B = electric cable to machine, power pack in machine
		T = central door	C = hydraulic drive from container directly into machine
		A = door above main drive or in top of pressure wall	E = electric cable from container directly into machine
			H = medium voltage supply to machine (> 1000V)

Herrenknecht AG
77963 Schwanau · Germany
Tel. +49 (7824) 302 923
Fax +49 (7824) 302 364
avn@herrenknecht.de

www.herrenknecht.com

HERRENKNECHT
Tunnelling Systems

ANEXO B. Listado de las funciones del software

El software de la interfaz gráfica de guiado está compuesto por treinta funciones **.m* que se encuentran adjuntas al presente Proyecto de Fin de Carrera en la carpeta donde se ha diseñado el software.

A continuación se va proceder a organizarlas por funcionalidad:

- Función de inicialización del sistema:
  - presentacion.m

- Función principal de la interfaz gráfica:
  - panel_total.m

- Función de representación en dos dimensiones:
  - representación_2d.m

- Función de representación en tres dimensiones:
  - representación_3d.m

- Funciones de cálculo de parámetros reales de la máquina tuneladora:
  - route_head.m
  - route_button.m
  - route_central.m

- Funciones de cálculo de parámetros simulados de la máquina tuneladora:
  - route_head_simulada.m
  - route_button_simulada.m
  - route_central_simulada.m

- Funciones de cálculo de parámetros reales de la flecha simbólica:
  - route_head_3d.m
  - route_button_3d.m
  - cruceta_negativa_x_3d.m
  - cruceta_negativa_y_3d.m
  - cruceta_positiva_x_3d.m
  - cruceta_positiva_y_3d.m

- Funciones de cálculo de parámetros simulados de la flecha simbólica:
  - route_head_3d_s.m
  - route_button_3d_s.m
  - cruceta_negativa_x_3d_s.m
  - cruceta_negativa_y_3d_s.m
  - cruceta_positiva_x_3d_s.m
  - cruceta_positiva_y_3d_s.m

- Funciones de representación de parámetros de la representación en 2D:
  - grafica_1cuadrante.m
  - grafica_2cuadrante.m
  - grafica_3cuadrante.m
  - grafica_4cuadrante.m

- Funciones de representación de parámetros de la representación en 3D:
  - grafica_1cuadrante_3d.m
  - grafica_2cuadrante_3d.m
  - grafica_3cuadrante_3d.m
  - grafica_4cuadrante_3d.m