

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSIDAD
POLITECNICA
DE VALENCIA



“Diseño y desarrollo de algoritmos de procesamiento para una estación ergonómica en una empresa del sector automovilístico.”

TRABAJO FINAL DE GRADO

Autor/a:

Vicent Avaria Avaria

Tutor/a:

José Ignacio Herranz Herruzo

Tutores/as empresa:

Vicente Puig Borrás

Marc Martínez Camarena

GANDIA, 2016

Resumen

En este proyecto se pretende realizar el diseño e implementación de un conjunto de algoritmos de procesado de información, para el cálculo de ángulos, distancias y su posterior representación, para acabar proyectando estos datos en un informe ergonómico *Sue Rodgers*. Este proyecto se encuentra integrado en un proyecto denominado Ergo Station, en el que utilizando el lenguaje de programación LabVIEW y cuatro cámaras KINECT, tratando de comprobar si las operaciones de una línea de montaje son adecuadas.

Palabras Clave: LabVIEW, KINECT, Algoritmo, Ergonomía, Procesado.

Abstract

This project aims to make the design and implementation of a set of a computer-process' algorithms for calculating angles, ranges and its representation, in order to project these data in a *Sue Rodgers* report. This project is part of a project known as Ergo Station, with the LabVIEW programing language and four KINECT cameras, which consists in checking if its assembly line's operations are appropriate.

Keywords: LabVIEW, KINECT, Algorithm, Ergonomics, computer-process.

I. INTRODUCCIÓN.	4
I.1. DESCRIPCIÓN DEL PROYECTO.	4
I.2. OBJETIVOS.	5
II. REQUERIMIENTOS NECESARIOS.	6
II.1. HARDWARE.	6
II.1.1. KINECT FOR WINDOWS V2.	6
II.1.2. SISTEMA COMPUTACIONAL DISTRIBUIDO	6
II.2. SOFTWARE.	8
II.2.1. LABVIEW 2015.	8
II.2.2. LIBRERÍAS.	9
II.3. ACCESORIOS Y ESPACIO.	12
III. ADQUISICIÓN DE DATOS.	14
III. 1. FUNCIONAMIENTO DE LA KINECT FOR WINDOWS V2.	14
III. 2. SISTEMA DE ADQUISICIÓN MULTI-KINECT.	16
IV. PROCESADO DE DATOS.	18
IV. 1. ALGORITMOS PARA EL CÁLCULO DE ÁNGULOS.	18
IV. 1. 1. PLANOS.	18
IV. 1. 2. CUELLO.	21
IV. 1. 3. ESPALDA.	22
IV. 1. 4. HOMBROS.	23
IV. 1. 5. CODOS.	24
IV. 1. 6. MUÑECAS.	24
IV. 1. 7. RODILLAS.	24
IV. 2. ALGORITMOS PARA EL CÁLCULO DE DISTANCIAS	25
IV. 3. ALGORITMOS PARA EL CÁLCULO DEL INFORME SUE RODGERS.	25
V. REPRESENTACIÓN DE DATOS.	29
V. 1. GRÁFICAS POR CICLO	29
V. 2. TABLAS INFORME SUE RODGERS	31
VI. INTERFAZ DE USUARIO.	32
VII. ANÁLISIS DE RESULTADOS.	35
VIII. CONCLUSIONES Y LÍNEAS FUTURAS.	37
IX. ANEXOS	38
X. BIBLIOGRAFÍA.	39

I. Introducción.

Hay diferentes análisis ergonómicos que se utilizan para analizar si los movimientos de un trabajador de cualquier empresa son nefastos para su salud. En concreto el análisis *Sue Rodgers*, trata de analizar tres componentes fundamentales: el esfuerzo, la duración y la frecuencia requerida por cada parte del cuerpo para realizar una determinada tarea. Por lo tanto, basándose en el cálculo de ángulos de las articulaciones y la distancia entre diversos puntos del cuerpo se pueden analizar estas tres componentes y elaborar así un informe Sue Rodgers.

En la actualidad, la mayoría de los procesos de análisis ergonómico *Sue Rodgers* en una empresa del sector automovilístico, se llevan a cabo por profesionales de la salud, que analizan en primera persona todos los matices, para poder realizar un estudio integral de clasificación y análisis de los puestos de trabajo. Surge pues un problema con las componentes objetivas del estudio *Sue Rodgers*, ya que estas podrían estar sometidas a errores debido a la subjetividad del profesional.

Por lo tanto, se puede afirmar que estos ángulos y distancias no se han calculado todo lo preciso que debería ser, ya que se introduce un error debido a que la evaluación se realiza con posturas estáticas, y además el hecho de que estos ángulos se realicen en diferentes planos al mismo tiempo, como el frontal o el sagital, añade otro grado de complejidad para el análisis. En consecuencia el cálculo de la duración de los esfuerzos y la frecuencia de estos puede verse alterado.

Otro problema de la metodología actual es que sólo se evalúan las posturas que el evaluador considera más críticas, pudiéndose equivocarse en esta valoración.

Actualmente, este problema se puede abarcar de diferentes maneras, debido al avance tecnológico que ha permitido la automatización de procesos, en concreto, y gracias a la aparición de la visión artificial, ha mejorado cualitativamente.

La visión artificial o visión computacional pretende dotar a los ordenadores de la capacidad para poder simular la visión humana, y así poder realizar un tratamiento, procesado y análisis de una imagen digital.

Mediante este proyecto se pretende conseguir una herramienta que pueda realizar un estudio ergonómico, sin la necesidad de que interfiera el factor humano en las componentes objetivas, en algunos de los puestos de trabajo de una empresa del sector automovilístico.

1.1. Descripción del proyecto.

Este trabajo se ha realizado enmarcado en la modalidad Prácticas en Empresa (SIE), por lo tanto esto implica que la empresa ha gestionado y supervisado todas las decisiones tomadas, y ha definido los requerimientos expuestos en el **apartado II**.

Por otra parte, y aunque el autor de este proyecto ha redactado todo el trabajo, gran parte del mismo ha sido realizado con un grupo de trabajo, y todos los miembros han aportado información para que la redacción de este sea óptima.

Aunque este trabajo haya sido gestionado por la empresa, ninguno de los puntos anteriores supone un problema o dificultad añadida al trabajo, sino que ofrecen una mayor definición de todos los puntos para poder llevar a cabo un trabajo más elaborado.

Por consiguiente, en el presente trabajo se documentan, en el **apartado II**, los requerimientos necesarios de hardware, de software, y otros aspectos a tener en cuenta, como los accesorios y espacio, para la realización de este proyecto.

A continuación, en el **apartado III**, se detallarán los métodos de adquisición de datos, analizando el funcionamiento de las cámaras Kinect V2 para poder comprender los algoritmos de procesado de datos del **apartado IV**.

Una vez se expliquen todos los algoritmos implementados, en el **apartado V**, se expondrán las diferentes maneras de representar los datos, y seguidamente, en el **apartado VI**, se presentará la interfaz de usuario, haciendo especial hincapié en los apartados donde se muestren los resultados de los algoritmos implementados. En el **apartado VII** se analizarán los resultados obtenidos. Para finalizar se expondrán las aplicaciones, líneas futuras y conclusiones, en los **apartados VIII y IX**.

1.2. Objetivos.

El **principal objetivo** de este trabajo final de grado es el diseño e implementación de un conjunto de algoritmos de procesado de información, para el cálculo de ángulos, distancias y su posterior representación. Enmarcado en un proyecto denominado Ergo Station que sea capaz de procesar todos los movimientos realizados en una línea de montaje del sector automovilístico por un operario.

De este objetivo principal, se desprenden los siguientes **objetivos secundarios**:

- Implementar algoritmos con el lenguaje de programación *LabVIEW* y cuatro cámaras KINECT de la forma más eficiente y robusta posible.
- Comprobar el correcto funcionamiento de los algoritmos implementados.

De manera que, en los **apartados IV, V y VI** se describen los algoritmos implementados para lograr llevar a cabo los objetivos de este proyecto. Pero el **apartado III** ha sido realizado en paralelo a este trabajo por parte del equipo Ergo Station. No obstante, debido a su importancia en el proyecto es descrito brevemente para la comprensión completa del trabajo.

II. Requerimientos necesarios.

II.1. Hardware.

A continuación se documenta el hardware utilizado para la realización de este proyecto, tanto para la adquisición de datos como para el procesado de los mismos

II.1.1. Kinect for Windows V2.

Las cámaras que se han utilizado para el proyecto son las cámaras *Kinect for Windows V2*, ya que se trata de un sensor de profundidad de bajo coste, que se adapta perfectamente a las exigencias del proyecto.

El número de cámaras variará según el puesto de trabajo, pero desde un principio se ha determinado que el número de cámaras son cuatro, y en el análisis de resultados siempre se han utilizado cuatro.

Estas cámaras *Kinect for Windows V2*, aportan unos requerimientos mínimos a nivel de hardware y también a nivel de software. Por la parte de hardware, es necesario un ordenador por cada Kinect, y además este habrá de tener puerto USB 3.0 como mínimo, así como también 4GB de memoria RAM y un procesador de 64-bit (x64), 3.10GHz o uno más rápido; además de una tarjeta gráfica que soporte DirectX11. Por la parte de software, será necesario un Sistema Operativo Windows 8 o superior.

Algunas de las características a tener en cuenta de la cámara *Kinect for Windows V2* (Figura 1):

- Campo de visión: 70° horizontal y 60° Vertical.
- Resolución máxima de 1920x1080.
- Rango de profundidad de 0,5m a 4,5m
- *Framerate* con el sensor de profundidad 30Hz (30 fps).
- 25 *joints* del esqueleto humano.



Figura 1: Cámara Kinect for Windows V2.

II.1.2. Sistema Computacional Distribuido

Como especifica la cámara *Kinect for Windows V2*, es necesario la utilización de un ordenador por cada Kinect que se utilice. Por esta razón se ha diseñado un sistema computacional distribuido, con cuatro ordenadores que hacen de clientes y uno que hace de servidor para almacenar todos los datos necesarios de cada cliente, y también

uno que utilice el HMI diseñado. Todos estos ordenadores se encuentran interconectados, creando así una red local con la ayuda de un *switch*.

Los cuatro ordenadores HP 260 G1 Mini (Figura 2) que hacen de clientes y uno que hace de HMI tienen las siguientes características:

- Intel Core i3
- USB 3.0 Port x 2, USB 2.0 Port x 2
- 4GB RAM



Figura 2: Ordenador Cliente HP 260 G1 Mini 4GB RAM, IntelCore i3.

El ordenador HP 260 G1 Mini (Figura 3) que hace de servidor, como debe de realizar funcionalidades que requieren más velocidad tiene las siguientes características:

- Intel® 5th
- USB 3.0 Port x 2, USB 2.0 Port x 2
- 8GB RAM



Figura 3: Ordenador servidor HP 260 G1 Mini 8GB RAM, IntelCore i5.

El Switch utilizado, TP_LINK TL-SG108 Switch (Figura 4), debe tener un mínimo de 6 puertos, pero se ha utilizado uno de 8.



Figura 4: TP_LINK TL-SG108 Switch.

II.2. Software.

En este apartado se presenta la información relativa al software utilizado para este proyecto. En el apartado anterior se ha especificado que las cámaras *Kinect for Windows V2* necesitan que el ordenador en cuestión tenga el sistema operativo Windows 8 o superior. Por otra parte, también es necesario que el servidor tenga el Microsoft Office 2013 o superior, y el *Adobe Reader* o cualquier programa que pueda visualizar archivos .pdf, para así poder visualizar los documentos generados y no tener ningún problema con los algoritmos.

II.2.1. LabVIEW 2015.

Dentro del apartado de Software, es importante remarcar el programa *LabVIEW 2015*, que es el que se ha elegido para diseñar y desarrollar tanto los algoritmos de procesamiento de datos, como para la adquisición y representación de estos en la interfaz, que también se ha implementado utilizando este.

LabVIEW es una plataforma y entorno de desarrollo que utiliza el lenguaje G (de programación visual gráfico). Por consiguiente, los algoritmos que se quieran implementar no se escriben, sino que se dibujan. Además, el hecho de tener un amplio grupo de librerías ya pre-diseñadas, facilita al usuario la creación de un proyecto, y de poder dedicar más tiempo a la interfaz gráfica.

El programa trabaja con archivos .vi, que constan de dos partes, el panel frontal, que es la interfaz donde el usuario puede ver los datos del programa actualizados en tiempo real, y el diagrama de bloques, donde se define la funcionalidad del programa, el código propiamente dicho, pero a base de bloques. Estos VIs trabajan con controles e indicadores, que son las entradas y salidas de los programas.

Además *LabVIEW* tiene una funcionalidad para poder ver el recorrido que hacen los datos por dentro del programa, y poder así detectar los fallos o las funcionalidades de VIs ya creados.

Para este proyecto, *LabVIEW* es la pieza angular, ya que gracias a su facilidad de programación gráfica, y su velocidad de procesamiento, se pueden implementar algoritmos más fácilmente, dándole a este proyecto un toque característico.

II.2.2. Librerías.

Anteriormente, se ha nombrado que *LabVIEW* trabaja con librerías pre-diseñadas, y en este apartado se detallan cada una de las utilizadas en este proyecto.

Primeramente, cabe destacar que *LabVIEW* 2015 ya tiene muchas librerías preestablecidas, que son parte del programa y están desarrolladas por *National Instruments*, como puede ser *Math Plots* o *Vision Development Module*, utilizadas en este proyecto.

Por otra parte, hay otras librerías necesarias para llevar a cabo este trabajo final de grado, y que a continuación se explican en detalle.

- *Haro3D*.

Haro3D es la librería encargada de dar acceso a algunas de las funcionalidades de la cámara *Kinect for Windows V2*, así como a las *Oculus Rift DK2*, no utilizadas en este caso.

En esta librería se pueden encontrar diferentes VIs, que controlan las funcionalidades necesarias para extraer los datos de la cámara *Kinect for Windows V2* (Figura 5).

- *Kinect2_AudioBeam_API.vi*
- *Kinect2_Body_API.vi*
- *Kinect2_Cloud_API.vi*
- *Kinect2_Depth_API.vi*
- *Kinect2_Fusion_API.vi*
- *Kinect2_Image_API.vi*
- *Kinect2_IR_API.vi*
- *Kinect2_Control_API.vi*

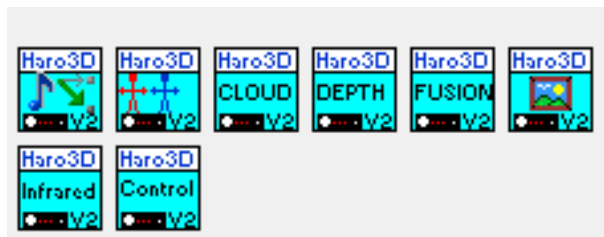


Figura 5: VIs de la Sub paleta *Kinect for Windows V2* de la librería *Haro3D*.

En este proyecto se utiliza el VI *Kinect2_Cloud_API.vi*, para obtener la nube de puntos de las *Kinect*, y así poder comprobar que el calibrado del sistema de adquisición *Multi-Kinect* (apartado III.2) es adecuado. El VI *Kinect2_Image_API.vi* se utiliza para obtener la imagen que captan las cámaras. Por otra parte, el VI *Kinect2_Control_API.vi* se utiliza para controlar el estatus de las cámaras en cada momento. Finalmente, el VI que se utiliza para la adquisición de los cuerpos para después poder hacer todos los cálculos de los algoritmos es el *Kinect2_Body_API.vi*, el cual tiene como salida un clúster con los 25 *joints* del cuerpo, e información necesaria sobre cada *joint*, como por ejemplo, si el *joint* se encuentra inferido o traqueado adecuadamente.

- *SQLite3*.

La librería *SQLite3* es la encargada de realizar las funcionalidades de la base de datos, pero sin la necesidad de crear ningún servidor. Al tratarse de una base de datos con archivos, es muy ligera. Esta base de datos se utiliza para almacenar todos los datos generados en la adquisición y en el procesado de datos, para que la aplicación de interfaz de usuario haga los mínimos cálculos posibles, y así poder obtener una aplicación más rápida y eficaz. (Figura 6)

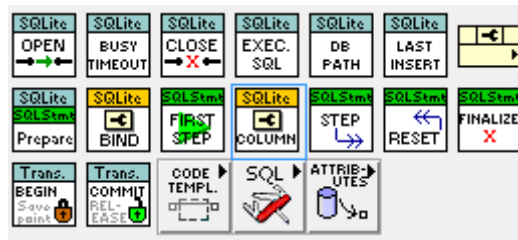


Figura 6: VIs de la Paleta de la librería SQLite3.

De todos los VIs que existen, tan solo se explicarán tres de ellos, ya que son los más utilizados durante todo el proyecto:



- SQLite Open: Es el VI encargado de abrir el archivo que contiene la base de datos para poder realizar todas las operaciones de insertado, actualizado o borrado.



- SQLite Close: Es el VI encargado de cerrar la base de datos, ya que si esta se quedase abierta, utilizaría parte de la memoria RAM y podría ser perjudicial para la aplicación.



- Execute SQL (Poly): Es el VI encargado de ejecutar las sentencias SQL que el usuario necesite en formato de texto (*String*). Entre las más usadas para el proyecto están las siguientes:
 - Create Table: Para crear una tabla donde después se almacenaran los datos.
 - Insert: Para insertar un grupo de datos en forma de filas en una tabla de la base de datos.
 - Delete: Para eliminar un grupo de datos en forma de filas en una tabla de la base de datos.
 - Update: Para actualizar un grupo de datos en forma de filas en una tabla de la base de datos.
 - Select: Para seleccionar un grupo de datos en forma de filas en una tabla de la base de datos.

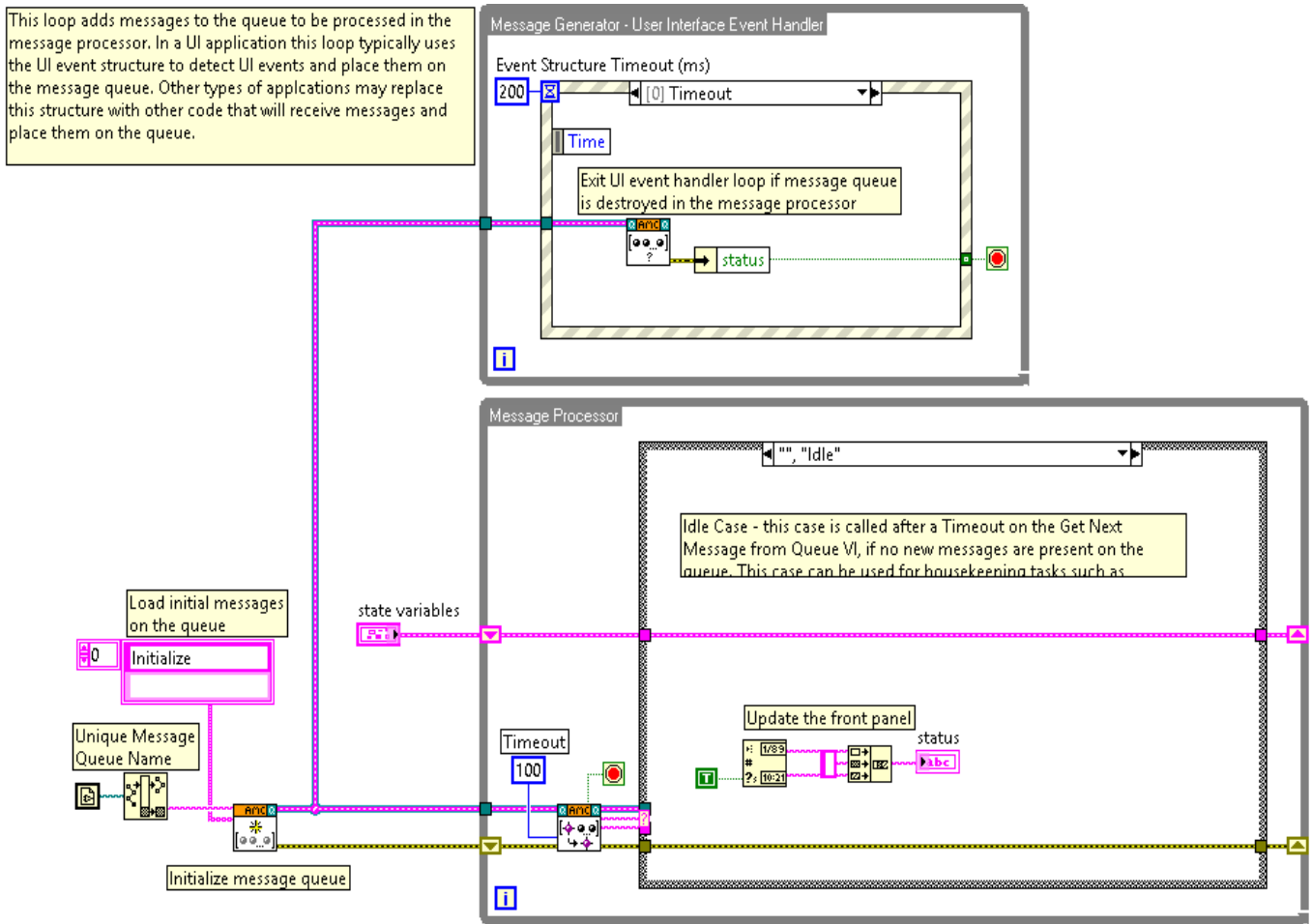
- AMC.

La librería AMC (The Asynchronous Message Communication) se utiliza para enviar mensajes en los procesos de los programas, entre procesos del mismo y entre diferentes tarjetas de *LabVIEW*. En este proyecto se utiliza para el control de los estados dentro de cada SubVI.

A diferencia de las librerías anteriores, esta no ofrece una paleta donde poder elegir VIs, sino unas estructuras predefinidas que realizan las funcionalidades de intercambio de mensajes.

En la figura 7 se puede observar un ejemplo de su *QMH Template.vi*, que se utiliza en todas las pestañas de la interfaz de usuario que se explica en el **apartado VI**.

Figura 7: Ejemplo de QMH Template.vi de la librería AMC.



-Modbus_TCP_AUTIS.

La librería Modbus_TCP_AUTIS no se encuentra publicada para que los usuarios la usen ya que ha sido diseñada por la empresa Autis Ingenieros S.L.U pero es la encargada de realizar las conexiones y peticiones entre los clientes y el servidor. (Figura 8)

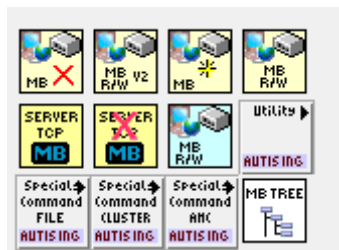


Figura 8: VIs de la Paleta de la librería Modbus_TCP_AUTIS.

Estos son los que han sido utilizados para este proyecto.

- MODBUS_TCP_CLOSE.vi
- Easy_MODBUS_TCP_Read_Write.vi
- Build Command MB message AMC.vi
- MODBUS TCP Server.vi

- MODBUS TCP Server Close.vi
- MODBUS TCP Open.vi
- MODBUS TCP Close.vi
- MB Read Function Data Cluster.vi
- MB Write Function Data Cluster.vi
- Build Command MB Download Cluster Changes.vi
- Build Command MB Upload Cluster Changes.vi

II.3. Accesorios y Espacio.

Finalmente, también hay que tener en cuenta los accesorios necesarios para poder llevar a cabo el proyecto.

Cada *Kinect*, y el marcador especificado en el Sistema de adquisición Multi-Kinect (**Apartado III. 2.**), necesita un trípode para poder mantenerse a una altura de 1,5 metros a 2 metros, y poder obtener así los datos adecuadamente. Se ha optado, por 5 trípodes Pie Profesional DynaSun (Figura 8), que cumplen las siguientes características:

- Altura Mínima 82cm - Altura Máxima 2m
- Peso 1040g
- Diámetro 23,55mm



Figura 9: Trípode Pie Profesional DynaSun.

El marcador utilizado en el Sistema de adquisición Multi-Kinect (**Apartado III. 2.**), se compone de una placa metálica tamaño hoja A3, con la posibilidad de atornillado para el trípode anteriormente descrito, con la zona baja de la placa metálica en posición vertical. También una pegatina tamaño hoja A3 en mate para evitar los reflejos de las luces, como se explica en el **apartado III. 2.**

Por otra parte, debido a que el espacio de trabajo adecuado para que las cámaras Kinect V2 trabajen debe estar entre 0,5 metros y 4,5 metros, el espacio también es un punto a tener en cuenta. Para que cuatro Kinects trabajen adecuadamente colocadas en los vértices de un cuadrado, debe haber como mínimo una superficie de 2 metros

II. Requerimientos necesarios.

cuadrados, contando que debe haber un operario en el centro. Aunque después el montaje en uno de los puestos de trabajo aumente considerablemente esta superficie, como mínimo deben ser 2 metros cuadrados, para poder hacer las comprobaciones oportunas.

III. Adquisición de datos.

En este apartado se analizará cómo se realiza la adquisición de datos desde las cámaras Kinect for Windows V2 y cómo funciona el Sistema de adquisición Multi-Kinect.

III. 1. Funcionamiento de la Kinect for Windows V2.

Para poder analizar debidamente la adquisición de datos primeramente se debe comprender el funcionamiento de las cámaras Kinect. Estas cámaras están dotadas de una cámara RGB, capaz de trabajar a 1920x1080, de tres emisores de infrarrojos y de un sensor de profundidad. También un array de micrófonos, pero en este proyecto por el momento no se ha hecho uso de ellos. (Figura 9)

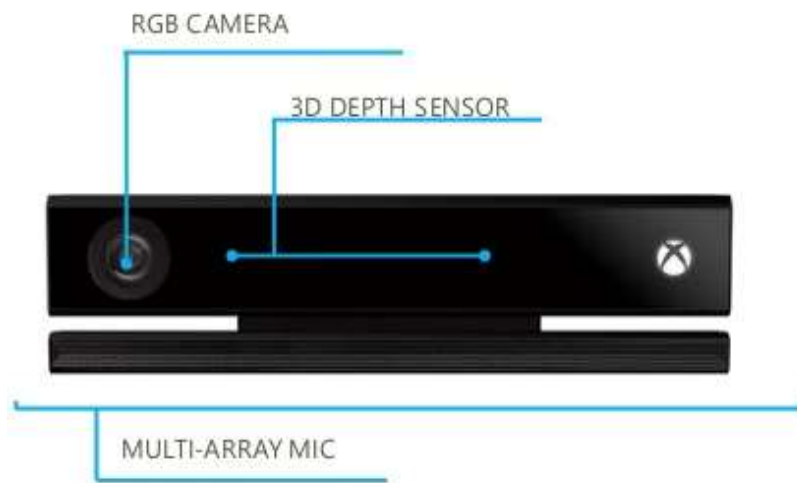


Figura 9: Distribución sensores Kinect for Windows V2.

La Kinect V2 trabaja en un sistema de coordenadas 3D, delimitando cada coordenada específicamente:

- El origen de coordenadas ($x = y = z = 0$) se encuentra en el sensor de profundidad.
- La coordenada x aumenta hacia la izquierda del sensor. (Rango: -1,5 a 1,5 metros)
- La coordenada y aumenta hacia arriba del sensor, teniendo en cuenta que hacia abajo se encuentra el soporte de esta. (Rango: -1 a 1 metro)
- La coordenada z aumenta en dirección hacia donde mira la cámara. (Rango 0,5 a 4 metros)
- Para la cámara Kinect la unidad es el metro.

La funcionalidad más característica de la Kinect es el algoritmo que utiliza para detectar el cuerpo humano y hacer un seguimiento de las partes detectadas. (Figura 10)

- 0 - SpineBase
- 1 - SpineMid
- 2 - Neck
- 3 - Head
- 4 - ShoulderLeft
- 5 - ElbowLeft
- 6 - WristLeft
- 7 - HandLeft
- 8 - ShoulderRight
- 9 - ElbowRight
- 10 - WristRight
- 11 - HandRight
- 12 - HipLeft
- 13 - KneeLeft
- 14 - AnkleLeft
- 15 - FootLeft
- 16 - HipRight
- 17 - KneeRight
- 18 - AnkleRight
- 19 - FootRight
- 20 - SpineShoulder
- 21 - HandTipLeft
- 22 - ThumbLeft
- 23 - HandTipRight
- 24 - ThumbRight

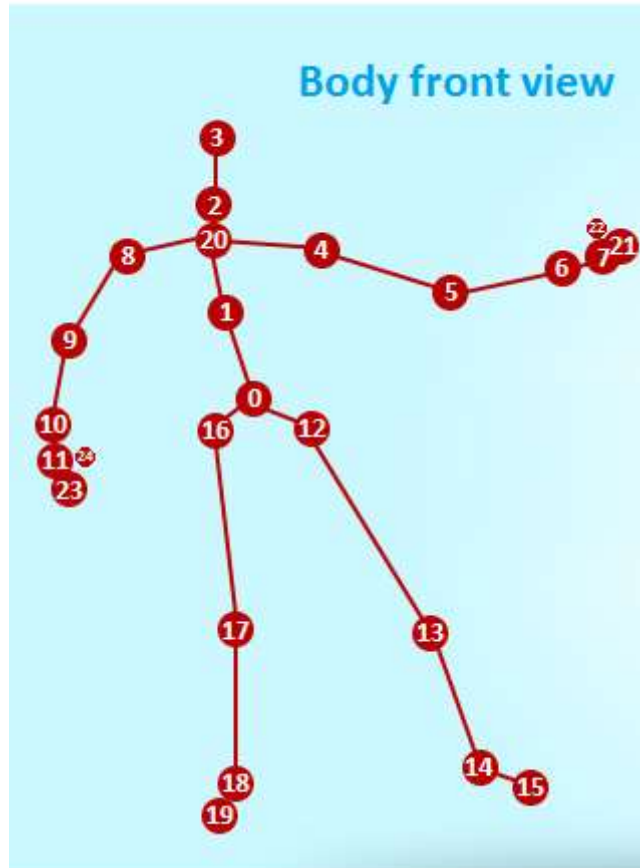


Figura 10: Joints detectados por la cámara Kinect for Windows V2.

A cada uno de estos Joints la cámara le asigna un valor 3D (x, y, z) de posición y cuatro valores de rotación de cada joint respecto de la cámara, pero en este proyecto tan solo se ha hecho uso de los valores 3D.

III. 2. Sistema de adquisición Multi-Kinect.

Una vez comprendido el funcionamiento de la cámara Kinect V2 el siguiente reto es construir un sistema de adquisición Multi-Kinect y poder obtener así unos datos más fiables.

Para eso, se tienen en cuenta diversos factores:

-Arquitectura del sistema.

Cliente-Servidor, cada cliente dispone de un Kinect v2 y envía el contenido a un ordenador servidor (Figura 11).

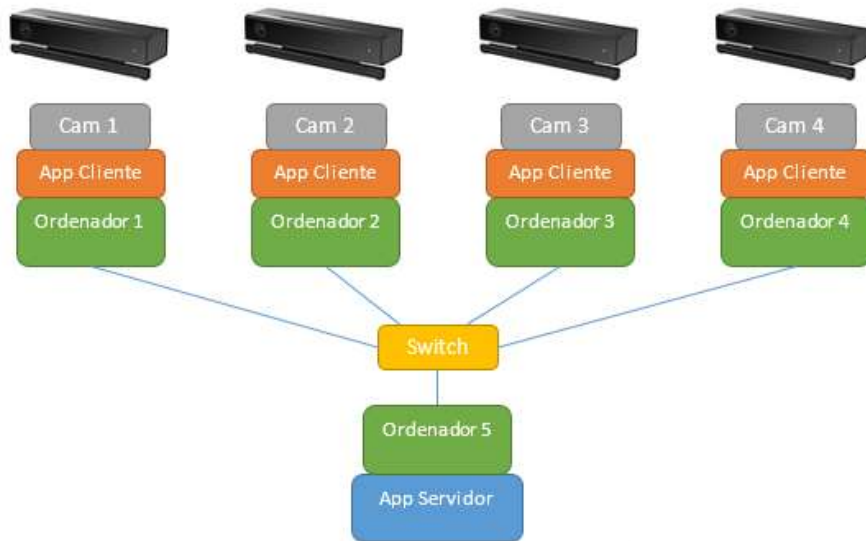


Figura 11: Arquitectura del sistema Cliente-Servidor.

-Sincronización Temporal.

Intercambio de mensaje eventos entre cliente y servidor para sincronizar a nivel de frame. El servidor estará conectado en modo escucha y será el que recibe las señales de los clientes que se conectan a él vía TCP/IP. También es el encargado de controlar a los clientes para así realizar una mejor calibración del sistema.

-Calibración.

Calibrado de los datos de todas las kinects clientes para crear un sistema de coordenadas global. De esta calibración se encargarán los clientes utilizando el marcador nombrado en el **apartado II.** (Figura 12)

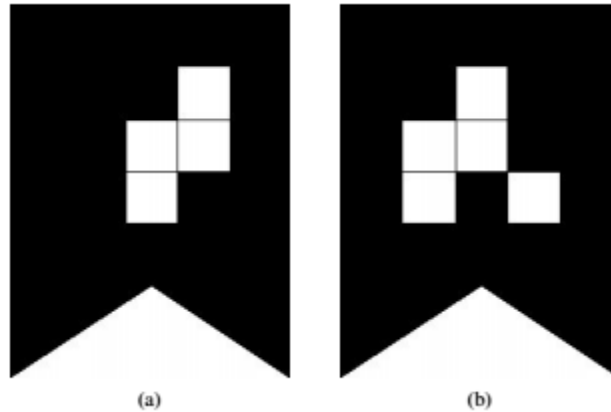


Figura 12: Marcadores para realizar la calibración en los clientes.

Para realizar esta calibración, los clientes dan una primera aproximación de la rotación y translación que tiene que dar cada nube de puntos proporcionada por cada cliente Kinect al sistema de referencia global que aporta el marcador en cuestión.

Se ha de tener en cuenta la posición en la que se encuentra el marcador, ya que según la posición en la que se encuentre los algoritmos de procesamiento de datos pueden variar, como se explica en el **apartado IV**.

-Fusion / Combinación Esqueleto.

Con los esqueletos calibrados de todas las kinects, los combinamos cogiendo la media de las articulaciones con estado de “tracking” bueno.

-Interferencias.

Está demostrado que puede haber interferencias entre Kinect cuando son utilizadas simultáneamente en sistemas múltiples debido a la tecnología ToF (Time of Flight) que utilizan para calcular la imagen de profundidad. Los estudios muestran que solo hay efectos negativos notables cuando las Kinect están mirando una superficie que refleja el haz luminoso de un dispositivo en dirección a otro dispositivo.

-Guardado de Datos en Cluster.

Guardado de los datos recibidos por cada Kinect cliente en el server de forma indexada en una carpeta compartida en red.

Este sistema de adquisición Multi-Kinect, como trabaja con cuatro cámaras sincronizadas, es inviable que consiga el framerate de 30Hz (30 fps) que se podía conseguir con una Kinect. Pero actualmente el sistema consigue trabajar a 10Hz (10 fps), siendo suficiente para no perder información sobre los movimientos de los operarios y obtener así un análisis más fiable.

IV. Procesado de datos.

En este apartado se analizarán todos los algoritmos diseñados e implementados para el procesado de datos. Se puede dividir en tres tipos de algoritmos diferenciados:

- El cálculo de ángulos, donde a través de cálculo vectorial sobre planos se obtienen los ángulos de cada articulación.
- El cálculo de distancias, donde se calculan las distancias entre las manos, entre los pies y también la distancia entre cada mano y su respectiva cadera para el plano transversal y el sagital, aportando así un análisis ergonómico más detallado.
- Cálculo del informe *Sue Rodgers*, donde se calculan los parámetros de esfuerzo, duración y frecuencia para cada parte del cuerpo.

IV. 1. Algoritmos para el cálculo de ángulos.

Se deben tener en cuenta diversos factores para el cálculo de ángulos, ya que con el sistema de adquisición Multi-Kinect se tiene un sistema de coordenadas delimitado por el marcador, descrito en el **apartado III. 2**. En consecuencia, el cálculo de ángulos no se puede realizar directamente con las coordenadas 3D de los joints adquiridos por el sistema Multi-Kinect.

IV. 1. 1. Planos.

Se definen diferentes planos biomecánicos que define el cuerpo humano, para así hacer un cálculo más adecuado de estos ángulos.

Los planos que se definen son los siguientes (Figura 13):

- **Plano Sagital**, el plano perpendicular a la línea recta que conecta la cadera derecha con la izquierda y pasa por el centro de la cadera.
- **Plano Frontal o Coronal**, el plano vertical que pasa a través de la cadera derecha e izquierda.
- **Plano Transversal**, plano horizontal que pasa a través de la cadera derecha e izquierda.
- **Plano Troncal**, plano determinado a través del cuello y la cadera derecha e izquierda.

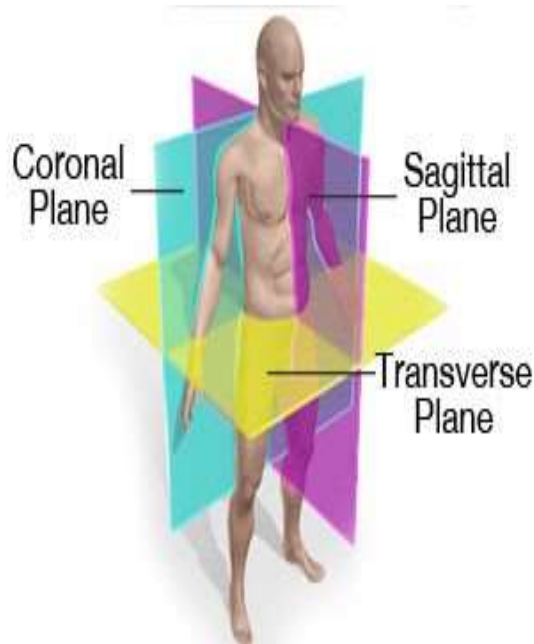


Figura 13: Planos biomecánicos definidos por el cuerpo humano.

Para el cálculo de cada uno de estos planos se ha descrito una ecuación a través de los joints adquiridos por el sistema Multi-Kinect.

Cada plano tiene una ecuación $Ax + By + Cz + D = 0$ donde A, B y C son el vector del plano el cual lo define. En cada plano, estos valores se han obtenido de una manera determinada, y se describen de la siguiente manera:

-Plano Sagital.

Para calcular el plano sagital, plano perpendicular a la línea recta que conecta el joint 16 'Hip Right' con el joint 12 'Hip Left' y pasa por el joint 0 'SpineBase' (Figura 10), la misma recta que definen los puntos de la cadera (izquierda y derecha) será el vector normal que define el plano.

Finalmente, para el cálculo del parámetro D, se escoge un joint cualquiera que pertenezca al plano, en este caso el joint 'SpineBase', y a través del producto escalar de este y el vector normal que define el plano se obtiene el parámetro D. (Figura 13)

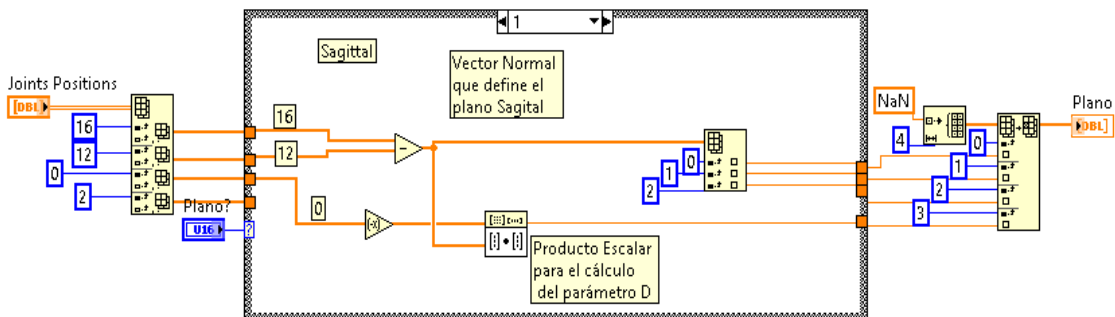


Figura 14: Código implementado para el cálculo del plano Sagital.

-Plano Frontal o Coronal.

Para calcular el plano frontal, plano vertical que pasa a través del joint 16 'Hip Right' y del joint 12 'Hip Left' (Figura 10), se define como vector normal del plano el producto vectorial entre la recta que une los joints 16 y 12, y la recta unitaria del eje de coordenadas 'y'.

Finalmente, para el cálculo del parámetro D, se escoge un joint que pertenezca al plano, en este caso debe ser el joint 16 o 12, y a través del producto escalar entre este y el vector normal del plano, se obtiene el parámetro D. (Figura 15)

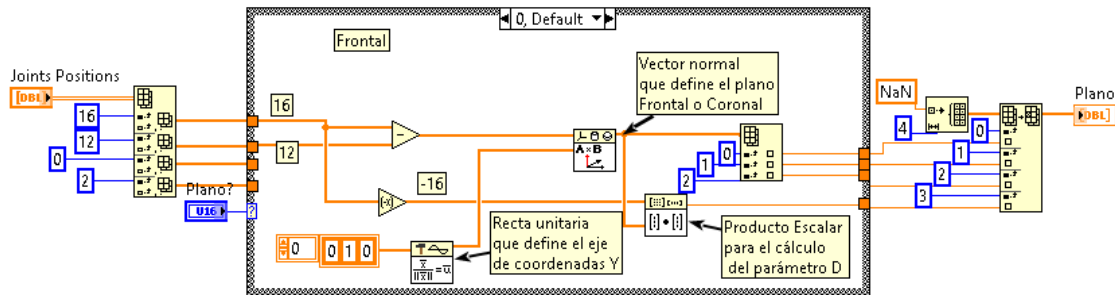


Figura 15: Código implementado para el cálculo del plano Frontal o Coronal

-Plano Trasversal.

Para calcular el plano transversal, plano horizontal que pasa a través del joint 16 'Hip Right' y del joint 12 'Hip Left' (Figura 10), se define como vector normal del plano el producto vectorial de la recta que une los joints 16 y 12, y la recta unitaria del eje de coordenadas 'X'.

Finalmente para el cálculo del parámetro D, se escoge un joint que pertenezca al plano, en este caso el joint 16 o 12, y a través del producto escalar entre este y el vector normal del plano, se obtiene el parámetro D. (Figura 16)

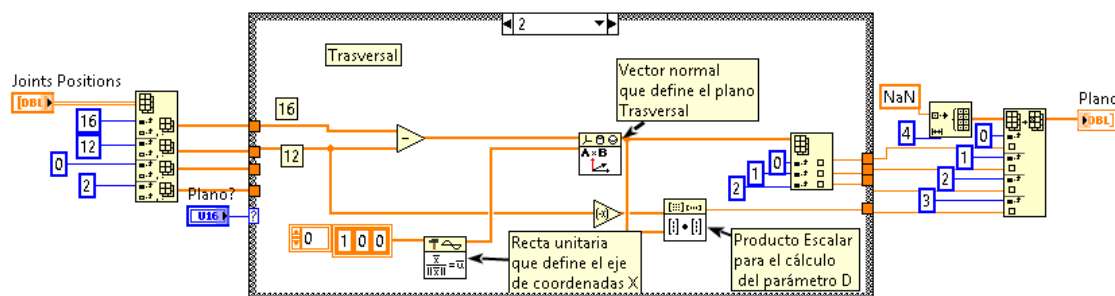


Figura 16: Código implementado para el cálculo del plano Trasversal.

-Plano Troncal.

Para calcular el plano Troncal, plano determinado a través del joint 2 'Neck' y los joints 16 'Hip Right' y 12 'Hip Left' (Figura 10), se define como vector normal del plano el producto vectorial de la recta que une los joints 16 y 12, y la recta que une el joint 2 y uno de los joints anteriores.

Finalmente para el cálculo del parámetro D, se escoge un joint que pertenezca al plano, en este caso el joint 16 o 12, y a través del producto escalar entre éste y el vector normal del plano, se obtiene el parámetro D. (Figura 17)

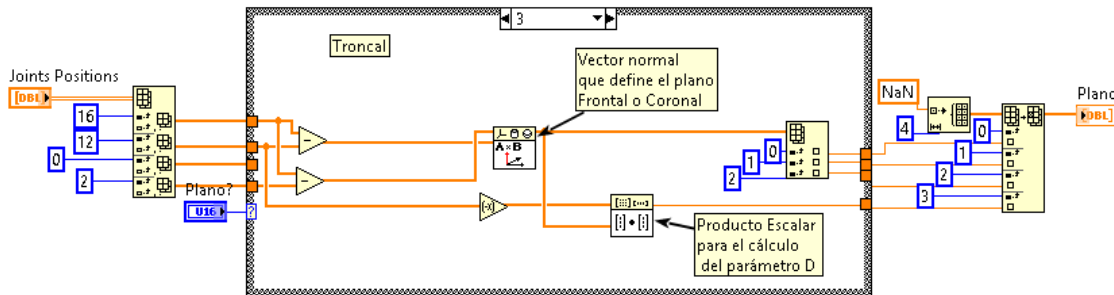


Figura 17: Código implementado para el cálculo del plano Troncal.

A partir de los planos definidos es necesario implementar un VI que proyecte los joints de interés para el cálculo de ángulos, que se define de la siguiente manera (Figura 18):

- Se construye un vector desde el joint de interés al plano en cuestión.
- Se calcula el producto escalar entre vector anterior y el vector normal del plano, obteniendo así la distancia ortonormal del joint respecto del plano.
- Multiplicando la distancia ortonormal por el vector de plano y a su vez restándosele al joint en cuestión se obtiene el joint proyectado sobre el plano en cuestión.

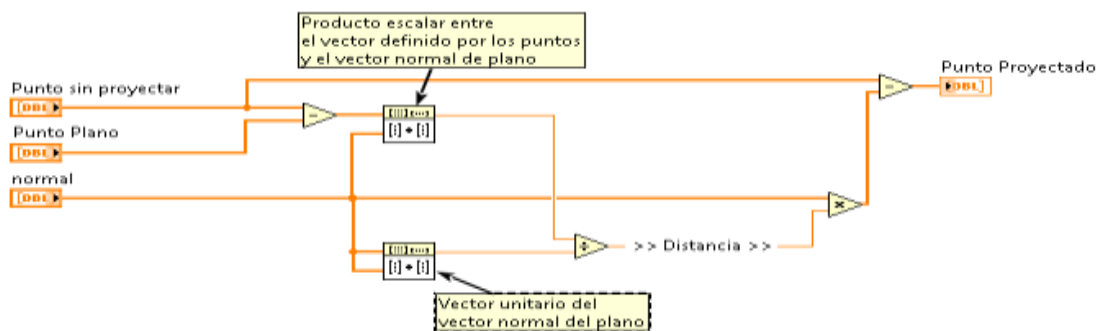


Figura 18: Código implementado para la proyección de un joint cualquiera sobre un plano cualquiera.

Una vez comprendido como se hace el cálculo de los planos y como se proyectan los joints sobre estos, se procede a la exposición de los algoritmos implementados para cada uno de los ángulos, agrupados por articulaciones.

IV. 1. 2. Cuello.

En el cuello se pueden diferenciar entre dos ángulos diferentes, el neck flexion/extension, que se calculan de la misma manera, y el neck lateral bend. Cada uno de ellos hace uso de un plano diferente. (Figura 19)

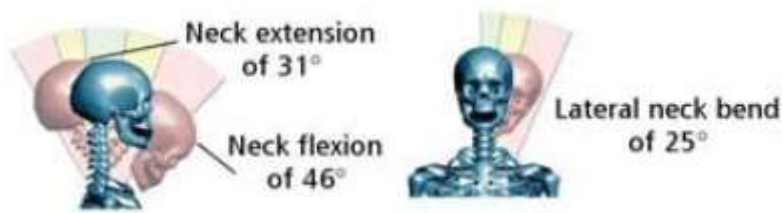


Figura 19: Ángulos calculados en el cuello, de izquierda a derecha, Neck extensión, Neck flexion y Lateral neck bend.

Para diferenciar entre el Neck flexion y el extension, se utiliza el plano sagital y se proyectan los joints 2 (Neck) y 3 (Head) sobre este. Una vez calculadas sus proyecciones se calcula el producto vectorial de las rectas que delimitan el ángulo, obteniendo así un vector normal. Según la dirección de este, se puede saber en qué cuadrante se encuentra el ángulo. Se negará posteriormente para poder utilizarlo después en su representación (**Apartado V**). Por lo tanto, la posición del marcador especificado en el sistema de adquisición Multi-Kinect es crucial para definir si se trata de Neck flexion o extension.

En la Figura 20 se puede observar cómo se ha implementado este algoritmo, donde se elige la componente X para diferenciar si se trata de flexion o extension porque el marcador se encuentra en el 2º cuadrante. Otra característica de este ángulo, es la necesidad de restarle a 180 el resultado para obtener un ángulo 0 en reposo.

A partir de ahora, todos los ángulos que necesiten comprobar la componente, tan solo se nombrará qué componente utiliza, pero no se analizará el código.

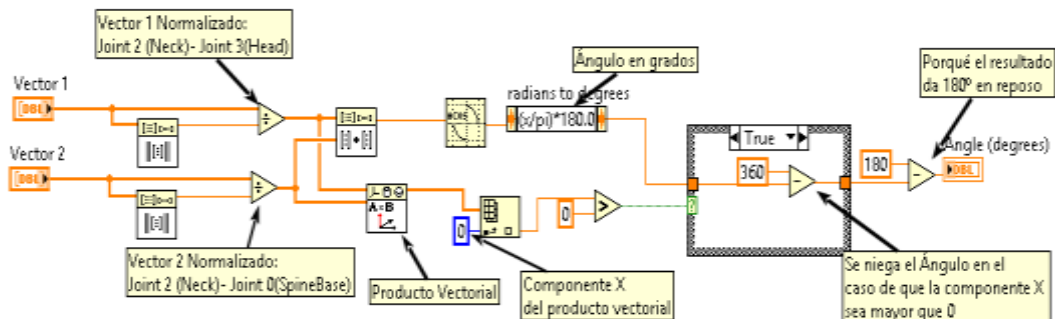


Figura 20: Código implementado para el cálculo de Neck Flexion y Extension.

El Neck Lateral Bend utiliza el plano frontal y la proyección del joint 2 y 3 sobre este, también se comprueba la componente del vector normal, la Z, para saber hacia qué lado se está moviendo y negar el ángulo para su posterior representación.

IV. 1. 3. Espalda.

En la espalda se pueden diferenciar cuatro ángulos, el back flexion y el extension, el back lateral bend y el back rotation. (Figura 21)

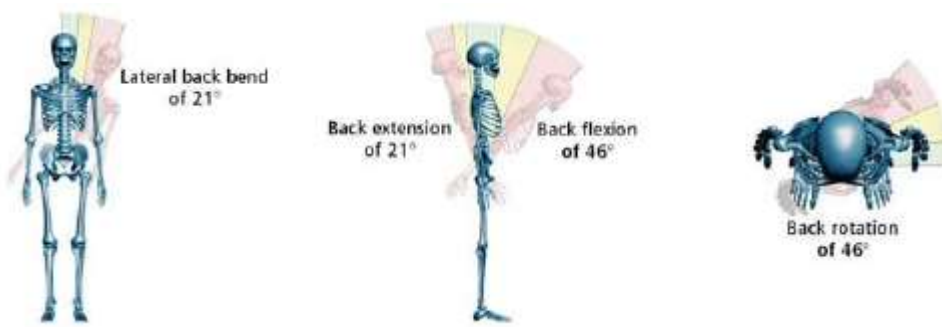


Figura 21: Ángulos calculados en la espalda, de izquierda a derecha Back Lateral Bend, Back Extension, Back Flexion y Back Rotation.

En referencia al back flexion y extension, se proyecta el joint 2 (Neck) sobre el plano sagital y se utiliza el vector unitario del eje de coordenadas 'Y' para realizar las medidas. En este caso también es necesario utilizar el cálculo del vector normal para saber si se trata de flexion o extension, utilizando la componente X, pero en este caso no es necesario restar el ángulo a 180 ya que en reposo es 0.

El ángulo back lateral bend, utiliza el joint 2 proyectado sobre el plano frontal y el vector unitario del eje de coordenadas 'Y' pero también se necesita comprobar la componente Z para saber hacia qué lado se está moviendo y negar el ángulo para su posterior representación.

Finalmente, el ángulo back rotation se calcula utilizando la recta entre los joints 4 y 8 (Shoulder Left y Shoulder Right) y también su proyección sobre el plano troncal. También es necesario hacer una distinción de hacia que lado se está moviendo y por lo tanto se comprueba la componente Z del vector normal. Se negará el ángulo para su posterior representación.

IV. 1. 4. Hombros.

En los hombros hay cuatro ángulos que deben ser calculados: Shoulder Flexion y Extension, Shoulder Abduction y Adduction, y cada uno de ellos para el hombro derecho y para el izquierdo. (Figura 22)



Figura 22: Ángulos calculados en los hombros, de izquierda a derecha, Shoulder Abduction y Adduction, Shoulder Extension y Flexion.

Para calcular el Shoulder Flexion y Extension, se proyectan los joints 5 y 9 (Elbow Left y Elbow Right) y también los joints 4 y 8 (Shoulder Right y Shoulder Left) sobre el plano sagital. Para diferencia entre flexion y extension también se utiliza la componente X del vector normal obtenido y el ángulo también se niega para su posterior representación.

Por otro lado, el Shoulder Abduction y Adduction necesitan la proyección de los joints 5, 9, 4 y 8 sobre el plano frontal. Se comprueba la componente Z del vector normal obtenido y se niega el ángulo para su posterior representación.

IV. 1. 5. Codos.

Los codos flexionados son uno de los ángulos que no necesitan calcular su producto vectorial entre vectores, ya que este se calcula a través de todas las componentes (3D). Tampoco necesitan restar 180 al ángulo obtenido, ya que su valor en reposo debe estar alrededor de 180°, pero se debe calcular para ambos codos. (Figura 23)

Por lo tanto con utilizar la siguiente formula, ya se obtendría el ángulo deseado:

$$Angle(degrees) = \left(\text{acos}(\widehat{V1} \cdot \widehat{V2}) * \frac{180}{\pi} \right)$$

Donde V1 es el vector entre los joints 4 y 5 o 8 y 9 (Izquierdo o derecho) y el V2 es el vector entre los joints 6 y 5 o 10 y 9 (Izquierdo o Derecho).

IV. 1. 6. Muñecas.

En el ángulo de las muñecas debería diferenciarse entre flexion y extension, pero debido a que no se puede calcular fácilmente un plano que delimite la mano, se ha optado por calcular el ángulo en 3D al igual que el de los codos, utilizando la misma fórmula, pero con los siguientes joints:

$$V1 = \text{Joint 11(Hand Right)} \\ - \text{Joint 10 (Wrist Right) o Joint 7(Hand Left) - Joint 6 (Wrist Left)}$$

$$V2 = \text{Joint 9(Elbow Right)} \\ - \text{Joint 10 (Wrist Right) o Joint 5(Elbow Left) - Joint 6 (Wrist Left)}$$

Pero es necesario restar el ángulo a 180, ya que la referencia de en reposo debe estar en torno a los 0°.

IV. 1. 7. Rodillas.

Finalmente, el ángulo de las rodillas flexionadas se ha calculado de la misma manera que los dos anteriores, utilizando la formula descrita en el apartado IV. 1. 5, pero utilizando los siguientes vectores:

$$V1 = \text{Joint 12(Hip Left)} \\ - \text{Joint 13 (Knee Left) o Joint 16(Hip Right) - Joint 17(Knee Right)}$$

$$V2 = \text{Joint 13(Knee Left)} \\ - \text{Joint 14(Ankle Left) o Joint 17(Knee Right) - Joint 18 (Ankle Right)}$$

En las rodillas, como en reposo el valor del ángulo ha de encontrarse alrededor de los 180°, no hace falta restárselo a 180.

IV. 2. Algoritmos para el cálculo de distancias

El cálculo de distancias se compone de cuatro tipos diferentes de medidas, la distancia entre manos, la distancia entre pies, la distancia desde cada mano hasta su cintura correspondiente en el plano trasversal y la distancia desde cada mano hasta el joint SpineBase.

-Distancia entre manos.

Para calcular la distancia entre manos se utilizan los joints 7 y 11 (Hand Left y Hand Right) y se calcula la distancia euclídea entre estos dos en centímetros. (Figura 23)

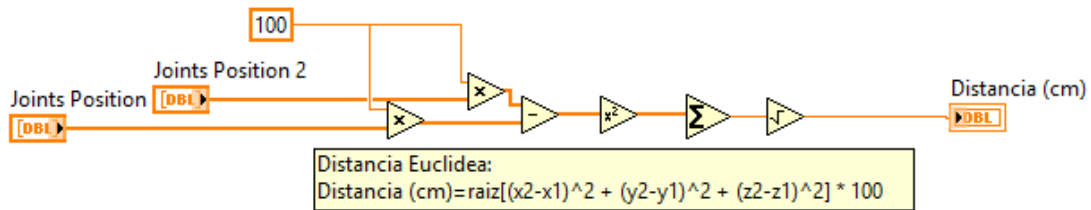


Figura 23: Código implementado para el cálculo de la distancia euclídea, en centímetros, entre dos joints.

-Distancia entre pies.

Para calcular la distancia entre pies se utilizan los joints 15 y 19 (Foot Left y Foot Right) y se calcula la distancia euclídea entre estos dos en centímetros, al igual que en la distancia entre manos.

-Distancia entre manos y cinturas, plano trasversal.

En referencia al cálculo de la distancia entre las manos (Joints 7 y 11) y su cintura correspondiente (Joints 12 y 16), no se calcula la distancia euclídea entre estos joints sino que los joints 7 y 11 se proyectan sobre el plano trasversal para más tarde calcular la distancia euclídea entre los joints proyectados y los joints de las cinturas.

En este apartado, también se guardan los joints 7 y 11 (Proyectados), 12 y 16 proyectados para después utilizarlos en un tipo de representación determinado que en el **apartado V** se analiza.

-Distancia entre manos y joint SpineBase, plano Sagital

Al igual que el apartado anterior, primero se proyectan los joints 7 y 11 sobre el plano sagital para después calcular la distancia euclídea entre estos y el joint 0 (SpineBase). En esta distancia también se guardan los joints 7 y 11 (Proyectados) y 0 para poder representarlos más tarde.

IV. 3. Algoritmos para el cálculo del Informe Sue Rodgers.

Para comprender debidamente el funcionamiento de los algoritmos se analizarán los componentes de este método de análisis.

El método *Sue Rodgers* analiza tres parámetros: esfuerzo, duración y frecuencia, como se había nombrado anteriormente, cada uno de ellos se evalúa individualmente, en una escala del 1 al 3, para cada parte del cuerpo. El grado de prioridad se determina a partir de la combinación de los valores asignados a cada parámetro (p. ej.: grado de prioridad =132, esfuerzo =1, duración =3, frecuencia =2).

-Nivel de esfuerzo

Los niveles de esfuerzo se valoran como ligeros (1), moderados (2) o fuertes (3) a través de una descripción cualitativa para las distintas partes del cuerpo. Como este proyecto trabaja a base de ángulos y distancias, pero aún no ha sido posible incorporar la componente fuerza, se han delimitado los niveles de esfuerzo a través de los siguientes valores (Tabla 1 y Tabla 2)

		Zonas de Movimiento (º)		
		1	2	3
Neck	<i>Flexion</i>	0 a 23	23 a 46	+46
	<i>Extension</i>	0 a 16	16 a 31	+31
	<i>Lateral Bend</i>	0 a 13	13 a 25	+25
Back	<i>Flexion</i>	0 a 11	11 a 46	+46
	<i>Extension</i>	0 a 11	11 a 21	+21
	<i>Lateral Bend</i>	0 a 11	11 a 21	+21
	<i>Rotation</i>	0 a 11	11 a 46	+46
Shoulder	<i>Flexion</i>	0 a 20	20 a 96	+96
	<i>Extension</i>	0 a 16	16 a 32	+32
	<i>Abduction</i>	0 a 35	35 a 68	+68
	<i>Adduction</i>	0 a 13	13 a 25	+25
	Elbow	180 a 120	120 a 100	-100
	Wrist	0 a 26	26 a 51	+51
	Knee	180 a 120	120 a 100	-100

Tabla 1: Zonas de movimiento o esfuerzo de los ángulos analizados

		Zonas de Movimiento (cm)		
		1	2	3
Entre	<i>Manos</i>	0 a 40	40 a 80	+80
	<i>Pies</i>	0 a 25	25 a 50	+50
Trasversal	<i>Mano a Cintura</i>	0 a 25	25 a 50	+50
Sagital	<i>Mano a Cintura</i>	0 a 25	25 a 50	+50

Tabla 2: Zonas de movimiento o esfuerzo de las distancias analizadas en centímetros

Algorítmicamente se diferencia entre zonas de esfuerzo con el simple hecho de que alguno de estos ángulos o distancias se encuentre enmarcado dentro de las zonas delimitadas.

-Duración del esfuerzo

La duración del esfuerzo es el tiempo que un músculo permanece activo de manera continuada durante un ciclo, en el caso de este proyecto, el tiempo que permanece un ángulo o una distancia en un determinado nivel de esfuerzo.

Se diferencian cuatro clasificaciones diferentes de 0 a 3, ya que puede haber zonas donde no haya duración alguna y por lo tanto es necesario valorar como 0 para no tener problemas en las representaciones. (**Apartado V**)

Para ángulos y distancias se utilizan las mismas clasificaciones (Tabla 3).

Clasificación	Duración del esfuerzo (s/ciclo)
0	0
1	0 a 6
2	6 a 20
3	+20

Tabla 3: Clasificación de la duración del esfuerzo para ángulos y distancias en segundos.

Para la implementación del código encargado de calcular la duración de cada zona hay que tener varios factores en cuenta:

- El framerate al que trabaja el sistema de adquisición, para así poder escalar a segundos el tiempo obtenido.
- Las zonas delimitadas por el nivel de esfuerzo. (Tabla 1 y 2)
- La clasificación de la duración del esfuerzo. (Tabla 3)
- Si el ángulo es positivo o negativo.

-Frecuencia

La frecuencia se mide para un grupo de músculos dado y para un nivel de esfuerzo específico, en el caso de este proyecto se mide las veces que un ángulo o distancia entra en un nivel de esfuerzo diferente. Debido a que es un trabajo muy repetitivo es necesario realizar un filtrado de suavizado para eliminar los picos que se crean por la desviación que incorpora el sistema de adquisición y así obtener un número de repeticiones más reducido, para que los valores del informe no se vean alterados.

Para ángulos y distancias se utilizan las mismas clasificaciones (Tabla 4).

Clasificación	Frecuencia/Ciclo
0	0
1	0 a 1
2	1 a 5
3	+5

Tabla 4: Clasificación de la frecuencia/ciclo para ángulos y distancias.

Para la implementación del código encargado de calcular la frecuencia/ciclo de cada zona, hay que tener varios factores en cuenta:

- Las zonas delimitadas por el nivel de esfuerzo. (Tabla 1 y 2)
- La clasificación de la frecuencia/ciclo. (Tabla 4)

-Prioridad

La prioridad de cambio se evalúa a través de los valores obtenidos de las componentes: Nivel de esfuerzo, Duración de esfuerzo y Frecuencia, en ese orden determinado para clasificarlo en cuatro valores: Nula, Baja, Moderada o Alta (Tabla 5).

Nula (Gris)	0 (100, 200, 300)
Baja (Verde)	1 (111, 112, 121), 2 (122, 131, 211, 212,113), 3 (221,311)
Moderada (Amarillo)	4 (123, 132), 5 (213, 222, 231), 6 (232, 312)
Alta(Rojo)	7 (223, 313), 8 (321, 322), 9 (323, 331), 10 (332), 11 (133, 233, 333)

Tabla 5: Clasificación de la prioridad para ángulos y distancias

Para la implementación algorítmica de la prioridad se deben tener en cuenta los valores de Nivel de esfuerzo (NE), Duración de esfuerzo (DE) y Frecuencia (E/M). (Figura 24)

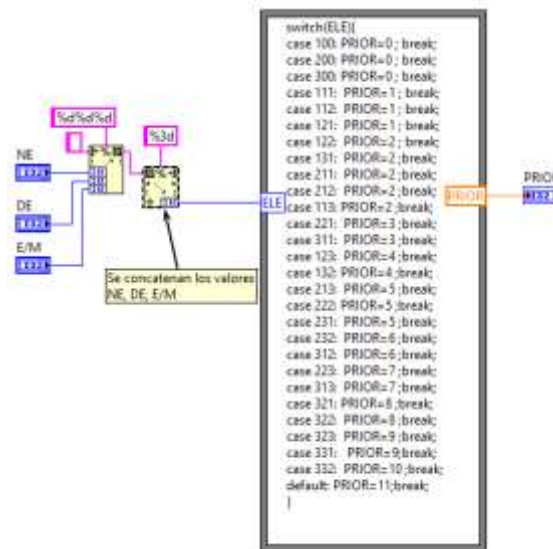


Figura 24: Código implementado para el cálculo de la prioridad.

V. Representación de datos.

En este apartado se analizan los métodos utilizados para la representación de los datos obtenidos con el procesamiento de datos (**Apartado IV**).

Se han implementado dos formas diferentes para representar los datos: las gráficas por ciclo de los ángulos y las distancias, y la representación en tablas del informe *Sue Rodgers* analizado en el **apartado IV. 3**.

V. 1. Gráficas por ciclo

Primeramente, decir que se ha escogido el VI *XYGraph.vi* para la representación de los ángulos y las distancias por ciclo, donde es necesario introducir los valores en el eje de abscisas (Tiempo segundos) y en el eje de ordenadas (Ángulos = Grados; Distancias = Centímetros). Por lo tanto, es necesario introducir un valor temporal a los valores obtenidos del procesamiento de datos. En la Figura 25 se puede observar como se ha implementado en LabVIEW, ya que es necesario saber el Framerate al que trabaja el sistema de adquisición y el número de valores que se quiere evaluar.

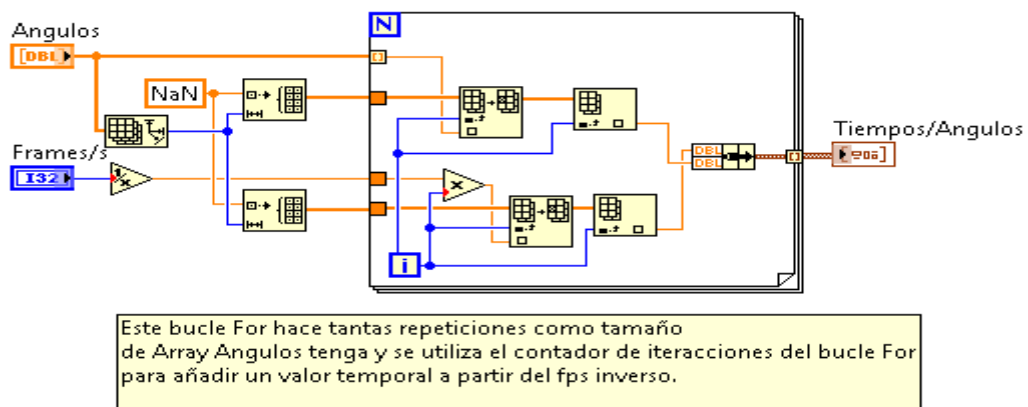


Figura 25: Código implementado para añadir la variable temporal a un array de ángulos o distancias determinado.

Después de obtener los valores para el eje X y para el eje Y, debemos configurar la gráfica, por tanto hay que tener en cuenta diferentes factores:

- La escala del eje de ordenadas (Tabla 1 y 2) más dos valores más que se añaden por arriba sumándole +20 al último valor. Se hace para que no desaparezcan valores que estén por encima de la zona 3.
- La escala del eje de abscisas, donde también se ha implementado un SubVI que se encargue de crear una escala con los valores múltiples de cinco hasta el valor del tiempo de ciclo indicado por el usuario (Figura 26)

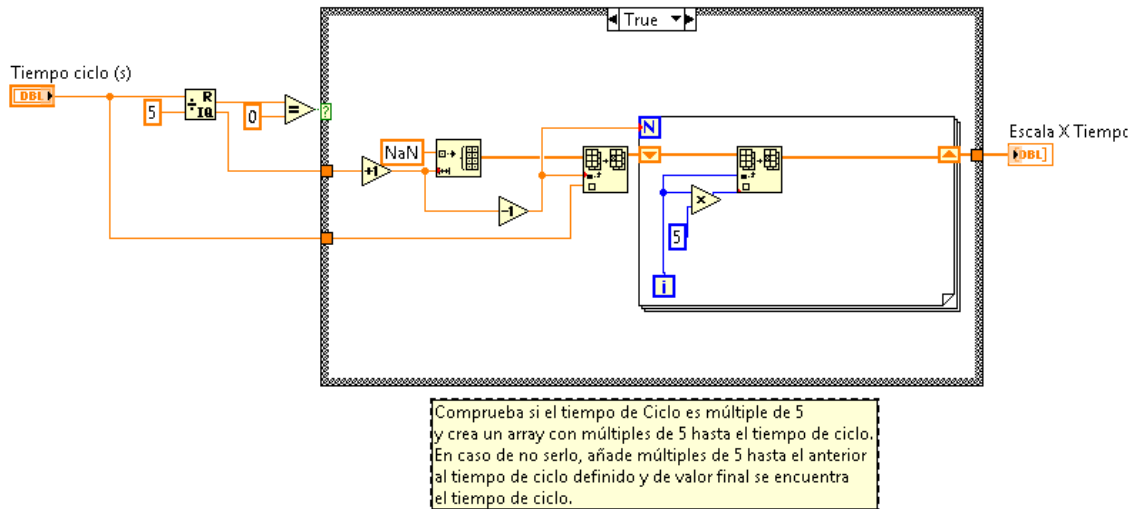


Figura 26: Código implementado para crear la escala temporal a partir del tiempo de ciclo.

Finalmente, se utiliza los *property node* del XY graph para configurarlo según los factores denominados anteriormente. (Figura 27)

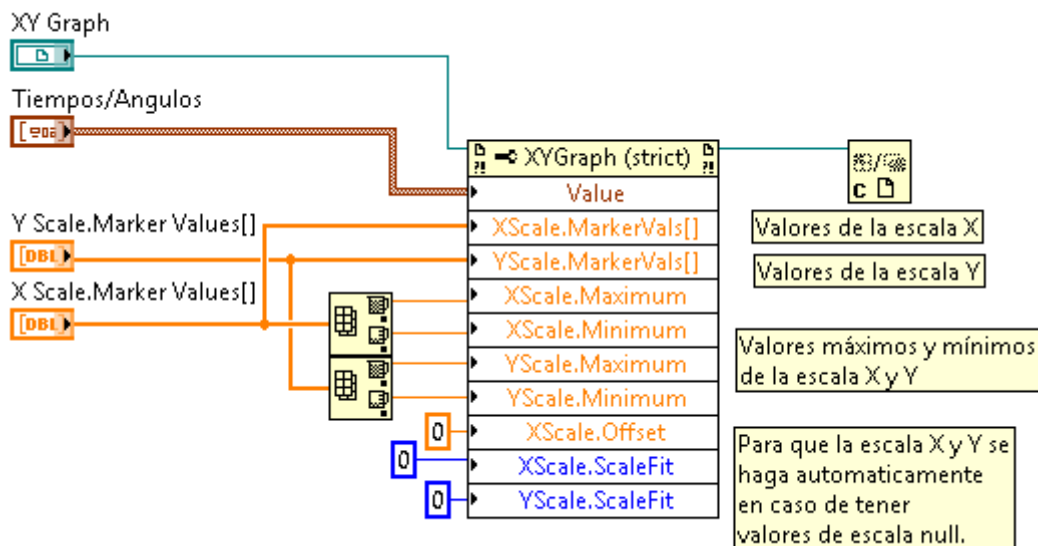


Figura 27: Configuración de la gráfica para ángulos y distancias según sus escalas y valores.

Obteniendo así un resultado parecido al de la figura 28, para el ángulo back flexion y extension.

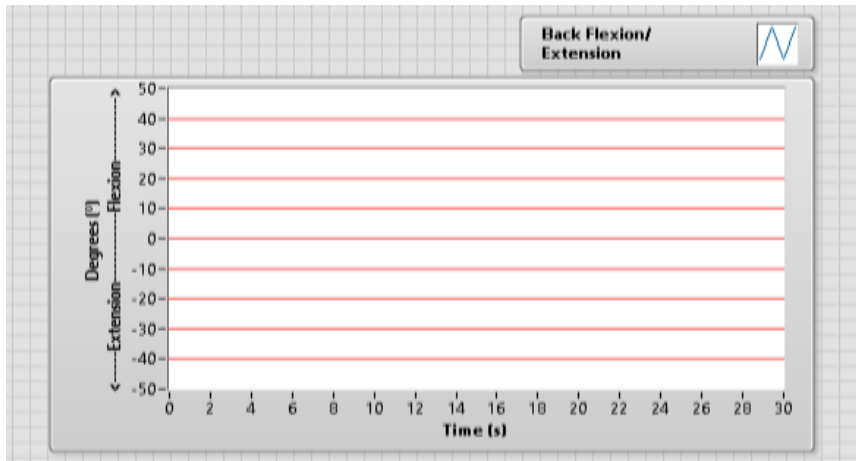


Figura 28: Ejemplo de gráfica para el ángulo Back Flexion/Extension, con tiempo de ciclo 30 segundos.

V. 2. Tablas informe Sue Rodgers

Para la representación de las tablas del informe *Sue Rodgers* se ha utilizado una plantilla denominada *Plantilla Informe.xlsx* que se encuentra en anexos debido a su extensión. Allí se representan las tablas con los valores de los parámetros de nivel de esfuerzo, duración del esfuerzo y la prioridad para cada uno de los ángulos y distancias calculados. Para ello se han utilizado los SubVIs de Active X, a través del software Excel 2013 (la Tabla 6 sirve de ejemplo para un ángulo)

Parte del Cuerpo	Nivel de Esfuerzo	Duración de Esfuerzo	Esfuerzos /Ciclo	Prioridad
Lateral Neck Bend	1	3	2	4
	2	1	1	2
	3	1	1	3

Tabla 6: Ejemplo de representación de las tablas *Sue Rodgers* para el ángulo Lateral Neck Bend.

Encontramos tres diferentes representaciones de estas tablas, la primera por zona de cada ángulo/distancia, que se mostraría como en la tabla anterior. La segunda por cada ángulo o distancia, guardando tan solo la zona (Nivel de Esfuerzo) que tenga la prioridad más alta. En el ejemplo anterior, el resultado para Lateral Neck Bend sería el siguiente (Tabla 7):

Parte del Cuerpo	Nivel de Esfuerzo	Duración de Esfuerzo	Esfuerzos /Ciclo	Prioridad
Lateral Neck Bend	1	3	2	4

Tabla 7: Ejemplo de representación de las tablas *Sue Rodgers* con el filtrado para el ángulo Lateral Neck Bend.

Finalmente, la última forma de representación sería por articulación evaluada, en el caso anterior sería el Neck (Cuello) y se escogería el ángulo/distancia (Lateral Neck Bend, Neck Flexion o Neck Extension) que tenga la prioridad más elevada.

VI. Interfaz de usuario.

En este apartado se analiza la interfaz de usuario diseñada para los apartados de representación (**Apartado V**) de los algoritmos implementados en el procesado de datos (**Apartado IV**). (Figura 29)

1. **Picture 3D:**

Se trata de la representación temporal del esqueleto unificado adquirido por el sistema de adquisición.

2. **Scrollbar Tiempo Actual:**

Se trata de un controlador de visualización del tiempo de ciclo que modifica el estado en el que se encuentra el esqueleto unificado. El Scrollbar se vuelve verde si se visualiza el esqueleto del operario y rojo si este no aparece en escena. También se ha implementado botones de play, pause o stop para poder controlar mejor la visualización.

3. **Scrollbars Movimiento 3D Esqueleto**

Se trata de tres scrollbars para mover el esqueleto en el eje X, Y y Z según convenga. Aunque el esqueleto se centra en el Picture 3D con la posición del joint 0 (SpineBase), puede haber problemas si aparecen más de un esqueleto y por lo tanto se puede activar o desactivar, con el boolean *Mover Esqueleto*

4. **Control Ciclo:**

Se trata de un control numérico de ciclos, que permite cambiar a cada uno de los ciclos del informe generado y modifica todos los valores de Graficas, Picture 3D, Duración de esfuerzo, Repeticiones y los valores de informe *Sue Rodgers*.

5. **Descartar Ciclo:**

Se trata de un control boolean que se utiliza para descartar aquellos ciclos que no sirven para el cálculo del informe *Sue Rodgers* debido a que el operario no está realizando la tarea, o no se encuentra en la escena o ha habido algún otro tipo de problema.

6. **Operario Disponible**

Se trata de un indicador boolean que se utiliza para marcar si el operario está o no disponible durante todo el ciclo. Si no se encuentra el operario se vuelve rojo y se marca como descartado para no hacer uso de él en el informe.

7. **Pestañas**

Se trata de un *tab control* de LabVIEW, donde te da a elegir que ángulo o distancia quieres visualizar, y modifica las gráficas de ángulos, el control de ciclos críticos, las duraciones de esfuerzo, las repeticiones y los valores del informe *Sue Rodgers*.

8. **Controles Ciclos Críticos**

Se trata de un control numérico con los ciclos críticos (Prioridad >7) de cada ángulo o distancia calculados. Cada ángulo/distancia tiene uno diferente, y con el cambio de este valor se modifica el control de ciclo, al ciclo crítico de ese ángulo/distancia.

9. Grafica Ángulos

Se trata de un XY graph, explicado su funcionamiento en el **apartado V. 1.** Contiene la información gráfica de cada ciclo y ángulo/distancia determinado.

10. Duración Esfuerzo

Se trata de un cluster con la información de la duración en segundos de cada zona y su porcentaje, tomando como total la suma de valor absoluto de todas las zonas del ángulo en cuestión.

11. Repeticiones

Se trata de una tabla con las repeticiones por ciclo que ha entrado en cada zona.

12. Valores Informe *Sue Rodgers*

Se trata de los valores de los parámetros: nivel de esfuerzo, duración del esfuerzo, repeticiones y prioridad del ciclo en cuestión para el ángulo elegido.

VI. Interfaz de usuario.

7- Pestañas de elección de ángulo que se quiere visualizar y la opción de ver la grabación de las cámaras.

9- Gráfica para visualizar el ángulo con su escala determinada y su tiempo de ciclo

10- Valores de duración en cada zona y el porcentaje total correspondiente de cada zona

11- Valores de las repeticiones por zona

1- Picture 3D: Representación Esqueleto (Joints).

2- Scrollbar: Posición temporal.

3- Scrollbars X Y Z: Para mover el esqueleto en 3D.

4- Control para elegir el ciclo que se quiere visualizar.

5- Botón para poder descartar un ciclo concreto.

6- Boolean para saber si hay o no operario en la grabación

8- Controles para elegir los ciclos críticos que se quieren visualizar

12- Valores calculados para el ángulo y ciclo concreto para realizar el informe Sue Rogers

Información Estadística Espalda

Back Lateral Bend | Back Flexion/Extension | Back Rotation | Cámaras

Gráfica Ángulo de un CICLO de Trabajo

Back Flexion/Extension

Gráfico: Degrees (°) vs Time (s)

TIEMPO por ZONA de MOVIMIENTO

Flexion

Zona	Porcentaje %	Valor Abs
Zona 1	100	17
Zona 2	0	0
Zona 3	0	0

Extension

Zona	Porcentaje %	Valor Abs
Zona 1	80	11
Zona 2	20	2.7
Zona 3	0	0

Repeticiones por ZONA de MOVIMIENTO

Flexion

Zona	Repeticiones
Zona 1	1
Zona 2	0
Zona 3	0

Extension

Zona	Repeticiones
Zona 1	5
Zona 2	2
Zona 3	0

Valores Analisis SUE RODGERS

	Nivel de Esfuerzo	Duración Esfuerzo	Esfuerzos /Ciclo	Prioridad
Flexion	Zona 1	2	2	2
	Zona 2	2	0	0
	Zona 3	3	0	0
Extension	Zona 1	1	3	4
	Zona 2	2	1	2
	Zona 3	3	0	0

Operario Disponible

Back extension | Back flexion

20/06/2016 09:23:55

Figura 29: Interfaz de usuario diseñada para el ángulo Back Flexion/Extension

VII. Análisis de resultados.

Para finalizar, una vez se ha analizado el funcionamiento de los algoritmos implementados y su representación. Se han hecho un total de tres pruebas en la empresa del sector automovilístico para poder ver de manera objetiva si todo funciona correctamente:

1ª Prueba 09/05/2016

En esta prueba se utilizaron cuatro cámaras Kinect, dos delante del operario y dos detrás. En ella se comprobó que las interferencias entre cámaras eran insignificantes y por lo tanto el sistema de adquisición Multi-Kinect funcionaba perfectamente.

También se comprobó la necesidad de eliminar la parte del esqueleto que se encuentra por debajo de la cintura de las cámaras delanteras, debido a que las cámaras delanteras no pueden recoger de manera adecuada estas partes y afectan negativamente a la fusión de los esqueletos, moviendo el esqueleto a una posición no adecuada.

En referencia a las cámaras traseras, se comprobó que las cámaras Kinect por mucho que el operario se encuentre de espaldas a las cámaras, éstas detectan como si el operario se encontrase de cara y por lo tanto los joints de los miembros se encuentran volteados hacia atrás y los de la parte izquierda del cuerpo se corresponden con la parte derecha del cuerpo. Para solucionar este problema, primeramente se define los joints del cuerpo izquierdo como los de la derecha y para evitar más problemas se eliminan los joints de los brazos, para así facilitar la fusión de esqueletos y poder obtener así un resultado adecuado.

En esta prueba aún no se encontraban los algoritmos desarrollados completamente, así como tampoco la interfaz (HMI). Se comprobó que los algoritmos de cálculo de ángulos estaban diseñados para poner el marcador delante del operario, y los resultados de los ángulos Flexion/Extension y Abduction/Adduction salían inversos, ya que el producto vectorial de estos vectores era de signo contrario al deseado. Por lo tanto, se definió que el marcador iba a estar detrás del operario y se modificaron los algoritmos de cálculo de ángulos.

Por otra parte, en esta prueba se comprobó la necesidad de creación de un filtro de suavizado, para los picos generados por el sistema de adquisición, ya que los resultados de frecuencia daban muy elevados.

El filtrado de suavizado implementado utiliza la siguiente fórmula:

$$Angulos\ Out\ [j] = \sum_{i=0}^{j=10} \frac{Ángulos\ Out[i]}{Orden}$$

Donde *Angulos Out*, son los valores obtenidos cada vez que se realiza el filtrado de un valor para obtener así un mayor efecto de filtrado de los valores.

2ª Prueba 02/06/2016

En esta prueba se utilizaron dos cámaras delante y dos detrás, ya se habían implementado los algoritmos de procesamiento de datos y había una primera versión de la interfaz (HMI).

Hubo también problemas con la posición del marcador ya que este se dispuso por detrás del operario, a la izquierda y a la derecha. Si se ponía a la derecha del operario los resultados volvían a salir inversos para los ángulos Flexion/Extension y

Abduction/Adduction. Se delimitó más la zona del marcador para acabar siendo por detrás y a la izquierda del operario.

Por otra parte, se observó la necesidad de incorporar a la interfaz el indicador boolean *Operario disponible* ya que había ciclos en los que el operario no se encontraba en la escena y deformaba el resultado del informe *Sue Rodgers*. Para ello se implementó una comprobación del informe resultante comparándolo con una constante que tuviese los mismos valores que un ciclo sin operario, y en caso de ser iguales éste se marcaba como que no había operario en la base de datos. También se valoró la necesidad del control de ciclos críticos para cada ángulo, para así hacer más fácil la búsqueda de los ciclos problemáticos.

Por otra parte, se comprobó que los frames de los videos y los ángulos calculados eran diferentes. El problema se hallaba en el insertado de valores Null a la base de datos, el cual se resolvió modificando estos valores Null por 0.

Con estos resultados, se descubrió que el filtrado de ángulos para el Lateral Back Bend, Lateral Neck Bend y Back Rotation no era suficiente. Debido a que, por ejemplo, los valores del Lateral Neck Bend en reposo daban alrededor de 20 grados, se optó por diferenciar hacia que lado se movía el cuerpo, para que así los valores negativos potenciasen el filtrado. Se optó por aplicar esta estrategia también en la representación de estos, para así obtener más información.

3ª Prueba 23/06/2016

En esta prueba tan solo se utilizaron dos cámaras delante y una detrás, y por tanto los resultados contenían menos información que las anteriores. Por otro lado, la interfaz HMI ya se encontraba en una fase muy avanzada de su implementación, y los resultados lo corroboraron.

Los algoritmos de procesado de datos no presentaron ningún error y tan solo se solicitó la posibilidad de incorporar un offset a los ángulos que el usuario piense que en reposo no son 0°, para así mejorar el resultado del informe *Sue Rodgers*. Se incluyeron también algunas mejoras a nivel de interfaz, como una guía para la puesta en marcha del HMI. Aparte de estas mejoras no se presentó ningún problema remarcable.

VIII. Conclusiones y líneas futuras.

Este trabajo se ha dedicado al diseño y desarrollo de algoritmos para el procesado de datos para una estación ergonómica en una empresa del sector automovilístico. Viendo los resultados obtenidos durante las tres pruebas realizadas *in situ* se puede valorar como satisfactorio el resultado de los algoritmos diseñados y desarrollados en *LabVIEW 2015*.

Los resultados obtenidos de los algoritmos implementados se pueden valorar satisfactoriamente debido a que estos son robustos y escalables, así como bien documentados, para que otro programador pueda hacer pequeñas modificaciones en ellos y mejorarlos para sus futuros usos.

Debido a la potencia de cálculo que aporta *LabVIEW 2015* los algoritmos implementados no aportan ningún problema para que el sistema de adquisición realice estos cálculos de ángulos y distancias en tiempo real mientras se captura, ya que en ningún momento estos superan los 100 milisegundos, aumentando así las prestaciones de los algoritmos.

Con el proyecto *ErgoStation* en su fase final, en el ámbito de procesado de datos no hay muchas mejoras que se pueden realizar y por tanto el papel de este trabajo dentro de este proyecto se ve finalizado.

Como líneas futuras, se debe decir que este proyecto, aunque se enmarque dentro de una empresa del sector automovilístico, sería fácilmente adaptable a cualquier trabajo que se lleve a cabo en cadena y se realicen movimientos repetitivos, así como también para el análisis ergonómico dentro de cualquier ámbito, ya sea deportivo o laboral.

Para concluir, solo cabe destacar el esfuerzo que se ha hecho desde la empresa, creando un grupo de trabajo profesional para llevar a cabo un proyecto que hubiese sido imposible de realizar desde la universidad y mucho menos posible por un alumno. Este trabajo ha favorecido el crecimiento y mejora, tanto a nivel técnico como a nivel personal de conocimientos y habilidades, por la realización de un proyecto real y tangible.

IX. Anexos

Los anexos están formados por la documentación del toolkit que se encuentra en un archivo denominado *Anexo Documentación Toolkit.pdf*, donde se nombran y describen todos los VIs implementados para la realización de este trabajo final de grado. También los componen unos ejemplos de los informes obtenidos denominados *Ejemplo informe 22-06-2016-12-07.xlsx*, *Ejemplo informe completo 22-06-2016-12-07.pdf*, *Ejemplo informe reducido angulos 22-06-2016-12-07.pdf* y *Ejemplo informe reducido articulaciones 22-06-2016-12-07.pdf*.

Debido a su amplia extensión estos se encuentran fuera de la memoria de este trabajo final de grado.

X. Bibliografía.

- BHANDARI, A. et al. (2014). "Resolving Multipath Interference in Kinect: An Inverse Problem Approach" en *IEEE Microsoft Research*.
- BONACCI, J. et al. (2014). "A General Pose Estimation Algorithm in A Multi-Kinect System" en *IEEE School of Engineering*, vol. 14.
- COLOMBO, G. et al. (2013). "Marketless Motion Capture Integrated with Human Modeling for Virtual Ergonomics". Milan: Department of Mechanical Engineering Polytechnic of Milan.
- CORTI, A. et al. (2016). "A metrological characterization of the Kinect V2 time-of-flight camera" en *Robotics and Autonomous Systems*, nº 72, p. 584-594.
- DELMAS, P. et al. (2013). Multi-Kinect Scene Reconstruction: Calibration and Depth Inconsistencies" en *International Conference on Image and Vision Computing New Zealand* (28. 2013. New Zealand). Auckland: University of Auckland.
- DELPRESTO, J. et al. (2013). "Safe lifting: An adaptive training system for factory workers using the Microsoft Kinect" en *Engineering Design Symposium* (2013. Charlottesville). Virginia: University of Virginia.
- DIEGO-MAS, J. A. Y ALCAIDE-MARZAL, J. (2014). "Using Kinect sensor in observational methods for assessing postures at work" en *Applied Ergonomics*, nº 45, p. 976-985.
- DORMANIC, I y TASHEV, I. (2014). "Hardware and algorithms for ultrasonic depth imaging". Redmond: Microsoft Research.
- GONZÁLEZ-JORGE, H. et al. (2015). "Metrological comparison between Kinect I and Kinect II sensors" en *Measurement*, nº 70, p. 21-26
- HAGGAG, H. et al. (2013). "Real-time ergonomic assessment for assembly operations using Kinect" en *International Conference on Computer Modelling and Simulation* (15.2013. Reino Unido). Australia: Deakin University.
- KHOSHELHAM, K. (2011). "Accuracy Analysis of Kinect Depth Data" en *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVIII-5/W12
- KLEINSORGE, M. et al (2014). "Ergo Assist: Am assistant system to maintain ergonomic guidelines at workplaces" en *IEEE Emerging Technology and Factory Automation*, vol. 14.
- KOWALSKI, M. et al. (2015). "LiveScan3D: A Fast and Inexpensive 3D Data Acquisition System for Multiple Kinect v2 Sensors" en *International Conference on 3D Vision* (2015. Warsaw) Warsaw: Warsaw University of Technology.
- LACHAT, E. et al. (2015). "First Experiencies with Kinect v2 sensor for close range 3D modelling" en *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol XL-5/W4.
- LEE, B. et al. (2014). "Kinect Skeleton Coordinate Calibration for Remote Physical Training" en *The International Conferences on Advance Multimedia* (6. 2014. Athlone). Athlone: Athlone Institute of Technology.
- MARTIN, C. et al (2012). "A real-time ergonomic monitoring system using the Microsoft Kinect" en *Engineering Design Symposium* (2012. Charlottesville). Viginia: University of Virginia.

- MCGORRY, E. et al. (2015). "The validity of the first and second generation Microsoft Kinect for identifying joint centre locations during static postures" en *Applied Ergonomics*, nº 49, p. 47-54.
- NEWCOMBE, R. et al. (2014). "KinectFusion: Real-Time Dense Surface Mapping and Tracking". Redmond: Microsoft Research.
- PHONSING, P. et al. (2015). "Multi-Kinect Cameras Setup for Skeleton based Action Recognition" en *International Conference of Information and Communication Technology for Embedded Systems* (6. 2015. Thailand). Hua Hin: IC-ICTES.
- PLANTARD, P. et al. (2015). "Pose estimation with a Kinect for Ergonomic Studies: Evaluation of the Accuracy using a Virtual Mannequin" en *Sensors*, vol. 15, p. 1785-1803.
- RODRÍGUEZ, A. y SEMPERE, F. (2003). "Clasificación de Puestos de Trabajo atendiendo a la fatiga muscular en una línea de montaje de automóviles" en *Congreso de Ingeniería de Organización* (5.2003.Valladolid-Burgos). Alcoy: Universitat Politècnica de València.
- SCHMITZ, A. et al. (2014). "Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system" en *Journal of Biomechanics*, nº 47, p. 587-591.
- SHOTTON, J. et al. (2011). "Real-Time Human Pose Recognition in Parts from single Depth Images". Cambridge: Microsoft Research Cambridge & Xbox Incubation.
- STARANOWICZ, A. et al. (2015). "Easy to use, general, and accurate Multi-Kinect calibration and its Application to Gait Monitoring for Fall Prediction".
- WANG, C. L. et al. (2015). "Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications". Hong Kong: The Chinese University of Hong Kong.
- YANG, L. et al. (2015). "Evaluating and Improving the Depth Accuracy of Kinect for Windows v2" en *IEEE Sensors Journal*, vol. 15, nº8.
- YEAGER, R. (2013). *An Automated Physiotherapy Exercise Generator*. Berkeley: University of California.