



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



Sensor inalámbrico de velocidad para el Ejercicio de Press de Banca mediante microcontrolador Arduino

MEMORIA PRESENTADA POR:

Cristian Menacho Anta

Máster Universitario en Ingeniería Mecatrónica

DIRECTOR:

Vicente Fermín Casanova Calvo

Septiembre 2017

Tabla de contenido

1.	Introducción.....	3
1.1	Objetivo.....	4
1.2	Modelos existentes en el mercado.....	4
1.2.1	Beast Sensor.....	4
1.2.2	Encoder lineal.....	4
1.2.3	Push band.....	5
1.3	Medidas significativas en el press de banca	5
1.3.1	Velocidad de ejecución	5
1.3.2	Inclinación	9
1.4	Medicina deportiva y salud.....	10
1.4.1	Lesiones en el powerlifting	10
1.4.2	Ejecución correcta del movimiento: pautas.....	11
2.	Estructura del prototipo	14
2.1	Equipo emisor	15
2.1.1	Microcontrolador Arduino Due	15
2.1.2	Batería sensores	16
2.1.3	WiFi Shield.....	17
2.2	Equipo receptor: Matlab & Simulink.....	19
2.3	Diseño del soporte mediante NX	23
3.	Resultados Experimentales.....	26
3.1	Ejecución incorrecta por inclinación lateral.....	27
3.2	Ejecución incorrecta por inclinación frontal	30
3.3	Ejecución incorrecta por perdida de velocidad.....	32
3.4	Ejecución correcta vs incorrecta por profundidad de repetición	34
3.5	Ejecución correcta	34
3.6	Imágenes varias de una prueba en tiempo real.....	37
4.	Conclusiones	39
4.1	Evaluación de los resultados y del global del proyecto	39
4.2	Mejoras posibles al proyecto en un futuro	39
5.	Presupuesto.....	41
6.	Pliego de Condiciones	42
	Objeto.....	42
	Especificaciones de materiales y equipos.....	42
	Ejecución y control de calidad	44

Pruebas de Servicio	44
7 Bibliografía.....	45
Anexo I Código Arduino IDE	46
Anexo II Código Matlab archivos y diagramas de bloques.....	50
Anexo III: Planos	57

1. Introducción

Hoy en día nadie duda de que tener un buen nivel físico aumenta la calidad de vida y la duración de la misma, es por eso que se está poniendo cada vez más de moda el entrenamiento con cargas. Probablemente el ejercicio más famoso en los gimnasios sea el Press de Banca, un ejercicio que consiste en realizar un empuje horizontal acostado sobre un banco mirando hacia arriba. Es un gran constructor de fuerza y masa muscular en los músculos que intervienen en la ejecución: pectoral, deltoides y tríceps. La técnica correcta no es complicada en absoluto, pero puede parecer difícil en personas no entrenadas o con problemas de movilidad que necesiten adaptación. Por eso, y porque es el ejercicio de empuje más famoso junto al Press por encima de la cabeza (OverHead Press), se ha considerado útil la monitorización de la velocidad de ejecución del ejercicio y si el movimiento es correcto y adecuado.

Por otro lado, se tiene en cuenta que es un movimiento de carácter competitivo y entrenado a nivel élite por varios deportistas como powerlifters, halterófilos, nadadores y utilizado en preparaciones de equipos de fútbol americano, rugby, baloncesto, balonmano y un largo etc... Por lo que el aumento del rendimiento en este ejercicio unido siempre a una buena ejecución del mismo puede reportar beneficios en el día a día en la modalidad concreta del deportista. Debido a ello, últimamente están saliendo al mercado algunos gadgets que permiten trackear la velocidad del levantamiento para dar una idea del porcentaje máximo levantado, mejora de fuerza, niveles de fatiga, tipos de fibras musculares del individuo y demás información de utilidad.



Ilustración 1 Press Banca de competición

Lo ideal sería conseguir monitorizarlo sin ningún cableado que moleste al atleta la ejecución del movimiento y que los resultados se vean de forma instantánea, o en su defecto, en pocos segundos a través de un terminal móvil. En este caso, se ha decidido evitar cableados pero la monitorización y seguimiento se va a realizar mediante un pc portátil. No es lo más óptimo, pero por falta de tiempo y complejidad asociada se ha decidido dejarlo así.

Los artilugios deportivos comerciales que muestran la velocidad de ejecución están destinados mayoritariamente al aumento del rendimiento de fuerza absoluta en el ejercicio, es decir, principalmente para powerlifters. El powerlifting es un ejercicio que consta de 3 movimientos, uno de los cuales es el press de banca; la puntuación total es el número de kg movidos en los 3 diferentes ejercicios.

1.1 Objetivo

El objetivo de este proyecto en definitiva es diseñar un prototipo de sensor comercial que pueda monitorizar las principales variables en el ejercicio del press de banca. Las variables son: la velocidad de ejecución y dos ángulos de desvío. Una vez conseguidos estos parámetros se podrán obtener conclusiones de la ejecución del ejercicio de carácter médico para evitar lesiones o enfocarlo a la vertiente deportiva de élite para la optimización del rendimiento en distintas modalidades donde el empuje vertical sea una parte importante del entrenamiento del atleta.

1.2 Modelos existentes en el mercado

1.2.1 Beast Sensor

Principalmente es una pulsera que se coloca en el antebrazo en la parte inferior del codo, aunque también se puede colocar en la barra del levantamiento, pero se necesita material extra para la adaptación. Tiene unas 8 horas de autonomía y sirve para contar las repeticiones y la velocidad que han sido llevadas a cabo. Pesa menos de 50 gr y tiene un software donde se indica algunos parámetros interesantes para el atleta.

Uno de los más completos del mercado, cumple con su cometido; como pega, a parte del precio elevado, la incomodidad de llevarlo en la zona del antebrazo y si la persona realiza mal la ejecución por falta de simetría no se nota en el análisis.



Ilustración 2:Beast sensor

1.2.2Encoder lineal

Otra forma de realizar lo mismo es un encoder digital unido a un hilo y crear una app; la principal ventaja es la simplicidad y la precisión. Lo malo es que no es inalámbrico y los que se venden son caros y algunos mal optimizados en cuanto a la frecuencia del sensor.



Ilustración 3:Encoder lineal

1.2.3 Push band

La mejor opción del mercado, parecida al Beast sensor en definitiva es una pulsera que se coloca con la luz apuntando arriba en la zona del antebrazo, con mucha más precisión, menos autonomía, pero más información complementaria externa de gran valor. Lo peor, su precio y los fallos de simetría que pueden provocar fallos en las palancas mal proporcionadas por el individuo.

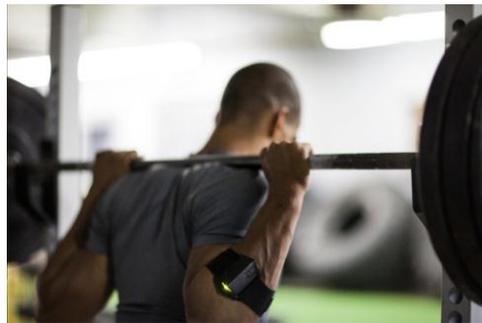


Ilustración 4:Push band

Como se ha visto, estos wereables son bastante interesantes en la práctica, pero tienen el problema del precio elevado, y de lo que puede ser un engorro: llevar sobre el brazo un elemento externo, a veces con luz, que pueden distraer e incomodar al sujeto que intenta mejorar. La precisión es buena, está claro que no es la misma que un encoder; pero el error está ahí y no es evitable.

1.3 Medidas significativas en el press de banca

1.3.1 Velocidad de ejecución

A la hora de intentar un levantamiento pesado pueden suceder varios problemas. El primero es que cuanto más pesado se levante, mayor riesgo potencial de lesión; la carga es una variable que hay que controlar para minimizar daños. El segundo es que los levantamientos potentes cercanos al fallo de forma frecuente fatigan sobremanera el sistema nervioso central, con lo que se perdería rendimiento. Pero existe una solución para saber si se han mejorado las

marcas del levantamiento pesado sin llegar a levantarlo, la solución es ver si la velocidad de ejecución se ha mejorado.

Como ejemplo se tiene que el día 1/1 un sujeto determinado levanta en su serie efectiva 85 kg por 5 repeticiones y con una velocidad máxima de repetición de 0.7 m/s. Ese mismo sujeto el día 5/5 se dispone a levantar el mismo peso que en aquella ocasión y lo hace a mayor velocidad, por ejemplo 0.85 m/s. Se puede decir que su habilidad y fuerza en el levantamiento ha sido superior que 5 meses atrás ya que como dice la intuición, una ganancia de velocidad es una ganancia de fuerza.

El ejercicio tiene dos fases, excéntrica y concéntrica. La primera es la fase donde se desciende la barra hasta apoyarla sobre el pectoral y la segunda es la fase propulsiva que se tiene que realizar con la mayor velocidad posible para estimular al máximo las ganancias de fuerza explosiva.

Es muy importante destacar que, cada persona tiene su propia velocidad de ejecución en cada levantamiento. Estas velocidades están determinadas por sus características individuales (experiencia en el ejercicio, morfología, predominancia de fibras rápidas o lentas etc.). Las velocidades extraídas de las tablas son una media de investigaciones realizadas en miles de personas, y aunque pueden ajustarse de forma bastante precisa, puede existir un pequeño error de estimación.

1.3.1.1 Aplicación BarSense Weightlifting

Esta aplicación que se encuentra en el Play Store de Google para dispositivos Android, y también en la App Store de iOS para dispositivos de Apple servirá de referencia externa al sensor diseñado en cuanto a la velocidad se refiere. La aplicación permite grabar el levantamiento, siempre de forma lateral y que los discos se vean perfectamente durante toda la trayectoria, y después da información en velocidad del levantamiento grabado indicando la velocidad estimada instantánea del recorrido.

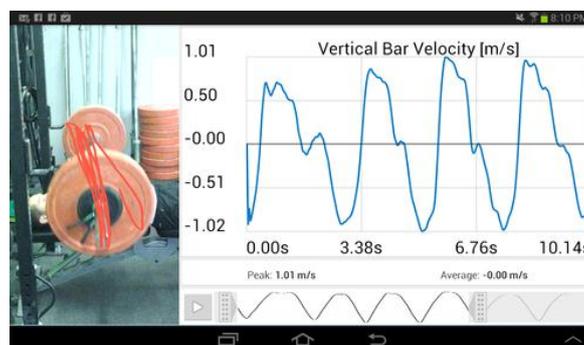


Ilustración 5: Barsense en Press Banca

1.3.1.2 Velocidad pico vs Velocidad media

Ahora se va a debatir la importancia del tipo de velocidad escogida para la verificación del levantamiento, a priori, se tienen dos velocidades: la velocidad máxima durante la repetición, la cual pasará a llamarse velocidad pico por temas de conveniencia según la literatura inglesa (peak velocity); y la velocidad media de la fase concéntrica.

La fuerza es el producto vectorial de masa y aceleración. La aceleración es la derivada de la velocidad. Por lo que la fuerza máxima se produce con la máxima aceleración, es decir, la máxima variación de velocidad por instante o periodo de muestreo. Según esto es lógico pensar que la única velocidad de utilidad es la velocidad pico. Nada más lejos de la realidad, es

tan o incluso más útil para este método la velocidad media; aunque la literatura científica nos dice que hay una pequeña ventaja a favor de la media si es tomada debidamente. El problema es esto último, aislar en los dispositivos la fase concéntrica sola y sacar una buena resolución hasta la finalización del ejercicio es tarea altamente complicada.

On the following table is the data for both Mean Velocity and Peak Velocity.

Lift	Load	Method	Trial 1	Trial 2	Trial 3	Mean	SD	CV%	Best Score
Bench Throw on Smith Machine (Pause 3" on chest)	20	Mean Velocity	1.25	1.25	1.24	1.247	0.005	0.38%	1.25
	40	Mean Velocity	0.99	0.94	0.94	0.957	0.024	2.46%	0.99
	60	Mean Velocity	0.71	0.68	0.67	0.687	0.017	2.48%	0.71
Parallel Squat (Pause 3" on the bench; no relax)	60	Mean Velocity	0.95	0.94	0.93	0.940	0.008	0.87%	0.95
	80	Mean Velocity	0.85	0.8	0.82	0.823	0.021	2.50%	0.85
	100	Mean Velocity	0.72	0.72	0.7	0.713	0.009	1.32%	0.72
Bench Press (Pause 3" on chest)	120	Mean Velocity	0.57	0.59	0.59	0.583	0.009	1.62%	0.59
	40	Mean Velocity	1	1.01	1.07	1.027	0.031	3.01%	1.07
	60	Mean Velocity	0.74	0.77	0.74	0.750	0.014	1.89%	0.77
Bench Press (Pause 3" on chest)	80	Mean Velocity	0.5	0.49	0.5	0.497	0.005	0.95%	0.5
	100	Mean Velocity	0.23	0.19	0.23	0.217	0.019	8.70%	0.23

Lift	Load	Method	Trial 1	Trial 2	Trial 3	Mean	SD	CV%	Best Score
Bench Throw on Smith Machine (Pause 3" on chest)	20	Peak Velocity	2.29	2.29	2.27	2.283	0.009	0.41%	2.29
	40	Peak Velocity	1.7	1.67	1.65	1.673	0.021	1.23%	1.7
	60	Peak Velocity	1.14	1.14	1.12	1.133	0.009	0.83%	1.14
Parallel Squat (Pause 3" on the bench; no relax)	60	Peak Velocity	1.69	1.7	1.71	1.700	0.008	0.48%	1.71
	80	Peak Velocity	1.56	1.51	1.52	1.530	0.022	1.41%	1.56
	100	Peak Velocity	1.39	1.36	1.34	1.363	0.021	1.51%	1.39
Bench Press (Pause 3" on chest)	120	Peak Velocity	1.15	1.21	1.21	1.190	0.028	2.38%	1.21
	40	Peak Velocity	1.63	1.73	1.74	1.700	0.050	2.92%	1.74
	60	Peak Velocity	1.18	1.19	1.17	1.180	0.008	0.69%	1.19
Bench Press (Pause 3" on chest)	80	Peak Velocity	0.78	0.66	0.75	0.730	0.051	6.98%	0.78
	100	Peak Velocity	0.4	0.28	0.44	0.373	0.068	18.21%	0.44

Ilustración 6: Velocidad media vs máxima

Viendo la tabla anterior, y tomando poniendo la mirada en la última prueba: el press banca con parada (quizás excesiva de 3 segundos), se observa que la covarianza es mayor en la velocidad pico que en la velocidad media. Esto no es un problema grande para tomas de máximas y relacionarlas con la carga de entrenamiento adecuado para cada sesión... En el proyecto la media se puede estimar a posteriori, pero la velocidad fija será clave y fácil de observar viendo la tabla de velocidad.

1.3.1.3 Mínima velocidad límite

Si se centra el foco en el press de banca competitivo, es decir, en levantamientos máximos de powerlifting hay varias cuestiones interesantes en cuanto a la velocidad en las que debemos ahondar. En primer lugar, decir que cada individuo tiene su propia curva de velocidad-peso levantado o %RM, es decir el porcentaje de la repetición máxima estimada. Una vez se tiene claro que un sujeto puede levantar el 90% RM más lento que otro su 100%; hay que mencionar un término que se usa para describir este tipo de levantamiento al límite o cerca del mismo.

El término mínima velocidad límite tiene relación con esa velocidad mínima necesaria para poder levantar la barra, esto tiene relación obviamente con un levantamiento máximo pesado. Y cada individuo tiene su propio límite de velocidad mínima, por lo tanto, hay que individualizar los estudios y resultados obtenidos de las lecturas del sensor; y saber así si ese sujeto ha progresado en cuanto a fuerza o se ha estancado, retrocedido... etc.

Force-Velocity Zones for Powerlifting

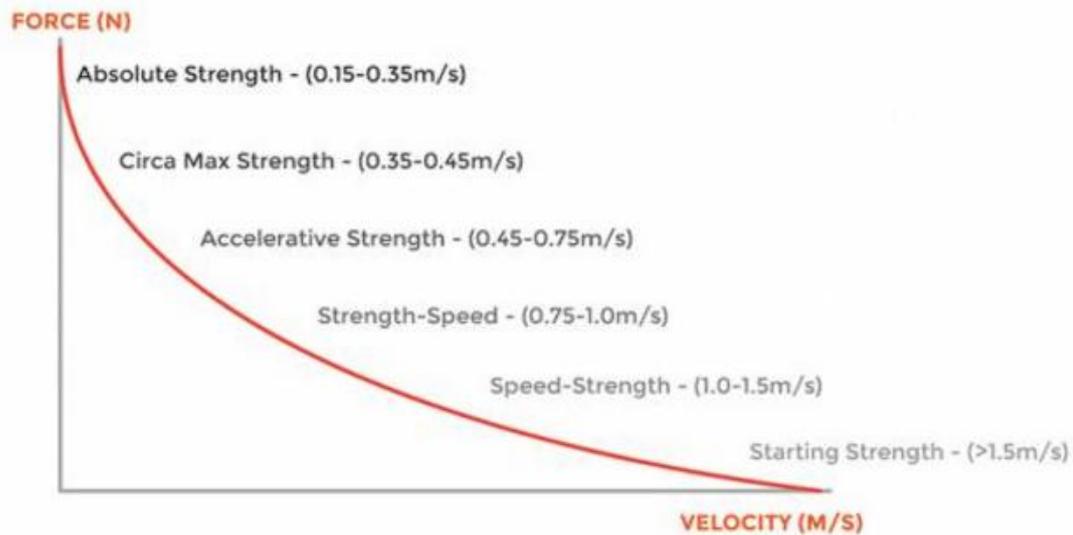


Ilustración 7: Zonas de trabajo de fuerza-velocidad

La imagen anterior debe servir de guía para ver las velocidades mínimas en cuanto a movimientos competitivos, como en este caso el press banca, para el press banca la mínima velocidad está en torno al 0.17 m/s de media y unos 0.35 m/s de máxima, pero depende del individuo esto puede variar.

1.3.1.4 Mediciones de velocidad con parada vs sin parada

Los últimos estudios indican que se al realizar la medición para la velocidad de ejecución en el press de banca, el hacer una pausa en el punto donde la barra está más baja, apoyada sobre el pectoral, supone un menor error y, por tanto, se obtiene datos de una fiabilidad más elevada que puedan ayudar a los expertos a una mejor interpretación de los hechos en sus enfoques deportivos. También añadir, que la parada es obligatoria en el press de banca de competición: powerlifting.

Ahora bien, si una parada aumenta la fiabilidad en las mediciones no quiere decir que todas nuestras series deben ser con parada para tener que evaluar siempre de la forma más fiable; pese al error se puede trabajar perfectamente sin parada y evaluar su rendimiento en un periodo de entrenamiento determinado sin problema, se notará perfectamente si hay un progreso notable o no. Pero para la competición y donde la velocidad es importante, siempre se recomienda con parada.

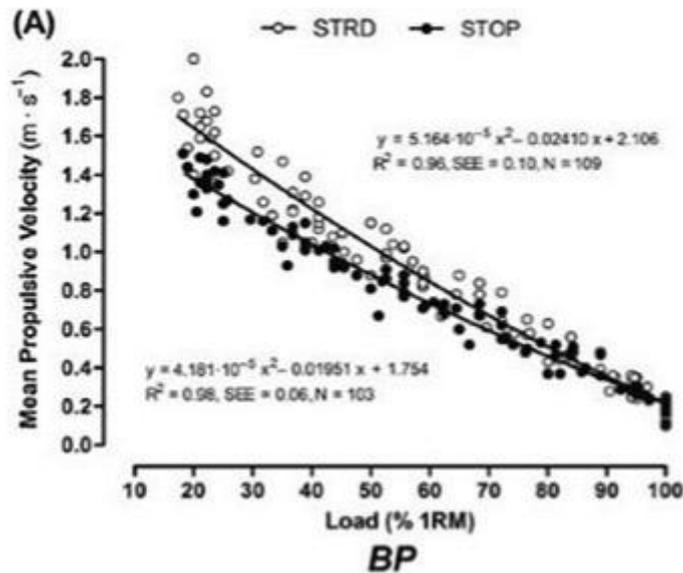


Ilustración 8: Pausa vs No pausa en medidas press banca

Como se aprecia en la figura, las medidas tomadas con pausa entre la fase de descenso y ascenso de la barra, es decir, con parada en el pectoral muestran una mejor correlación y menor desviación, con lo que se puede afirmar que son medidas más fiables.

1.3.2 Inclinación

El otro gran indicador de si la ejecución está siendo correcta, junto con la ya mencionada velocidad de ejecución durante el movimiento, es la inclinación sobre el eje de la barra, es decir si la barra va se inclina más de un lado que de otro. Además, también se mide el ángulo de la barra sobre su eje, es decir, si las muñecas rotan hacia adelante o hacia atrás en el ejercicio. Esta última permitirá saber si se produce algo suele ser un error grave a evitar en personas novatas y puede acarrear distintos problemas en la zona de la muñeca por la mala compactación de las articulaciones y la ineficiencia a la hora de distribuir el peso.

Dicho lo cual, se ha decidido implementar en la interfaz principal dos barras que indican en código de colores si la ejecución es correcta o no en tiempo real. Además, cuando finalice la ejecución saltarán por pantallas dos gráficas sobre la evolución de los respectivos ángulos respecto al tiempo. Así se podrían sacar conclusiones para ver si el paciente tiene alguna deficiencia unilateral, si existe alguna dolencia en las muñecas e intentar corregirlo.



Ilustración 9: Desbalance en el press de banca. Ángulo peligroso

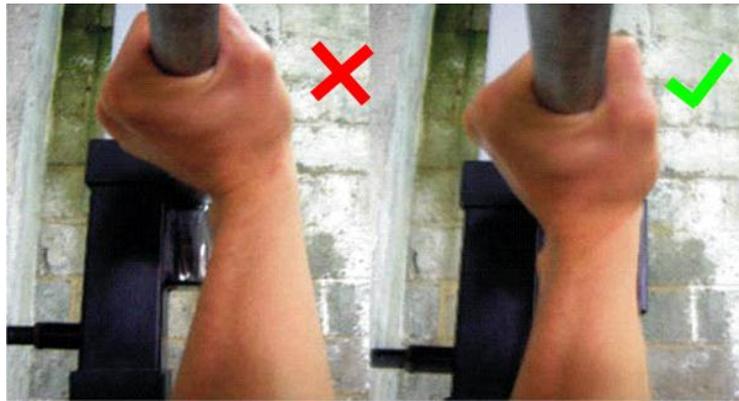


Ilustración 10: Error evitable giro de muñeca

1.4 Medicina deportiva y salud

1.4.1 Lesiones en el powerlifting

Existe riesgos de lesiones en la práctica del movimiento, estas lesiones son lesiones de sobreuso o crónicas, donde la repetición del ejercicio con una carga submáxima daña el sistema musculoesquelético, o bien cuando el descanso no es el necesario para permitir una adaptación estructural, apareciendo las conocidas tendinitis u otros dolores musculares. Es decir, las lesiones crónicas representan un tercio de las lesiones asociadas al levantamiento de fuerza; dicho lo cual en diferentes estudios exponen que el responsable de que esto suceda pueda ser una técnica pobre a la hora de levantar el ejercicio.

Siguiendo con el tema de las lesiones, se va ahondar las zonas más peligrosas de lesión en deportes de fuerza, en este caso las zonas más comunes de sufrir una lesión son la rodilla y el hombro. Este último es de gran importancia, puesto que el ejercicio del press de banca es un ejercicio multiarticular que sí implica esta articulación: la zona donde más se sufre el dolor si la ejecución en el empuje horizontal no es la adecuada.

Estudio	Deporte	Lesion por 1000 h	Lesiones más comunes
Winwood	Strongman	5,5	Espalda baja
Keogh (2006)	Powerlifting	4,4	Hombro
Quinney (1997)	Powerlifting	3,7	Espalda baja
Calhoun (1999)	Halterofilia	3,3	Espalda baja
Hak(2013)	Crossfit	3,1	Hombro
Raske and Norlin (2000)	Halterofilia	2,6	Espalda baja
Raske and Norlin (2000)	Powerlifting	2,6	Hombro
Haykowsky (1999)	Powerlifting	1,1	Espalda baja + Hombro
Siewe(2011)	Powerlifting	1	Hombro
Eberhardt (2007)	Culturismo	1	No especificado
Brown (1986)	Powerlifting	0,84	Hombro
Siewe(2014)	Culturismo	0,24	Hombro

Ilustración 11: Ratio lesiones donde se utiliza press banca u otro empuje multiarticular

En la gráfica se ven diferentes casos de atletas que utilizan el press de banca y demás ejercicios en su rutina específica del deporte que sea. Una de las zonas más comunes como se ha comentado en los levantamientos de fuerza es el hombro. (1) Como comentario adicional, hay que decir que los deportes de fuerza tienen, por lo general, menos ratio de lesión que los deportes convencionales como se vio en el estudio de Hamill en 1994.

Ratio lesiones en deportes (Hamill 1994)	
Deporte	
Futbol (escolar)	6,2
Rugby Gran Bretaña	1,92
Baloncesto USA	0,03
Cross Gran Bretaña	0,37
Squash	0,1
Futbol Americano	0,1
Badminton	0,05
Gimnasia USA	0,044
Powerlifting USA	0,027
Voleibol USA	0,0013
Tenis USA	0,001
Entrenamiento con pesas	0,0035
Halterofilia	0,00173

Ilustración 12 Ratio de lesión por 100 horas (2)

Aun así, cada persona es un mundo, las palancas no son exactamente igual para todos, el agarre biacromial no es el más cómodo para todos, unos se encuentran más cómodos abriendo más: en teoría sufre más la articulación del hombro, otros abriendo menos: recae más trabajo en la zona del codo... Por lo tanto, es importante los estudios biomecánicos que ayuden a entender ciertos patrones y fuerzas en los ejercicios, como por ejemplo palancas, momentos de fuerzas, ángulos... Esto sería de gran ayuda para entender los mecanismos del movimiento durante la lesión e intentar evitarlos o prevenirlos.

Otro tipo de lesiones en estos movimientos explosivos en ejercicios de fuerza son las lesiones traumáticas o agudas. Donde las lesiones son fácilmente detectables, es decir, se nota el momento justo que donde se produce la lesión. Por ejemplo, una contusión, un esguince de tobillo corriendo o una fractura de un hueso. En este tipo de lesiones, las traumáticas, el foco está puesto sobre los tendones o ligamentos, otra vez asociados a un movimiento inadecuado, poco seguro o carga excesiva.

1.4.2 Ejecución correcta del movimiento: pautas

En primer lugar, antes de intentar levantamientos pesados se debería intentar tener una técnica lo más correcta posible. Para ello se van a dar unas claves de lo importante que es la seguridad en este ejercicio con el fin de evitar lesiones tales como de hombro, de codo, de muñeca...etc. Estas lesiones principalmente son articulares, ya que es difícil tener una lesión muscular sin ir al fallo y sin ningún problema previo. El eslabón más débil y por tanto el que hay que fijar mejor y pretender que no sufra es la articulación y el tejido tendinoso.

Para una “técnica perfecta” en la ejecución del movimiento se debería tener siempre el glúteo pegado al banco durante toda la serie, sin despegarlo, a veces puede resultar complicado si se va al límite, pero nuestra seguridad y columna está en juego.

En segundo lugar, intentar que las muñecas no estén ni dobladas ni en extensión si no que formen una continuación del antebrazo, así se evitarán problemas en la articulación. También es importante que se peguen las escápulas con una retracción hacia atrás y hacia abajo en el banco. Es importante ya que así el movimiento se dará de forma más segura para los hombros intentando evitar la rotación interna de ellos.

A continuación, se debería fijar una posición de pies en la cual se esté cómodo, pero siempre apoyando las plantas en la superficie, es decir, sin despegar los pies; debería ser importante que no se deslicen e intentar siempre empujar los pies hacia abajo y hacia las puntas.



Ilustración 13 Press banca técnica segura

Volviendo al trabajo, para el sensor no se pueden medir o monitorizar todas estas pautas o consejos para la realización del movimiento, por tanto, se ha decidido simplificar un poco el tema de las lesiones; bien es cierto que las personas deberían tener un mínimo de técnica antes de realizar estos levantamientos. Además, en teoría la aplicación está pensada para dos vertientes: la competitiva donde tienen todos una gran técnica y las lesiones son minimizadas al máximo, y la vertiente de la rehabilitación en personas con problemas físicos que contarían con la ayuda de un profesional. Dicho lo anterior, las variables que se han considerado interesantes a la hora de evaluar la seguridad y riesgo del ejercicio a través del sensor son dos ángulos. Con uno se medirá si existe algún desbalance o descompensación entre palanca izquierda y palanca derecha, es decir, si existe un desnivel de fuerzas y se suba la barra más de un lado que del otro. Y el otro es el ángulo de rotación sobre las muñecas, es decir, si la barra no rota sobre su eje y por tanto no existe ninguna fuerza de rotación sobre la articulación muñeca.

Hay que tener claro que, pese a una buena técnica, un buen descanso del cuerpo donde esté recuperado totalmente en una sesión posterior se puede tener una lesión. Probablemente 2/3 lesiones de press banca tengan relación con el hombro. Y el hombro es la articulación más difícil de rehabilitar, esto también sirve para las personas que tienen poca movilidad y entonces el sensor diseñado se podría utilizar para monitorizar si el movimiento está siendo bueno una vez la rehabilitación esté en marcha, o incluso adaptarlo en diferentes situaciones y ejercicios, no todo va a ser press de banca. El hombro es la articulación que más movilidad puede tener y la que más rápido se posiciona. En el caso concreto del press de banca lo que

puede ocurrir que el húmero roce con la escápula y la clavícula, por eso, la gran importancia de la retracción escapular en este movimiento.

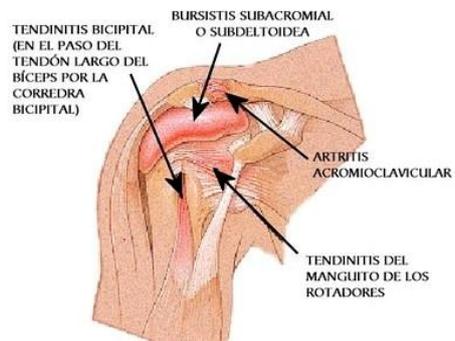


Ilustración 14 Diferentes patologías de la articulación

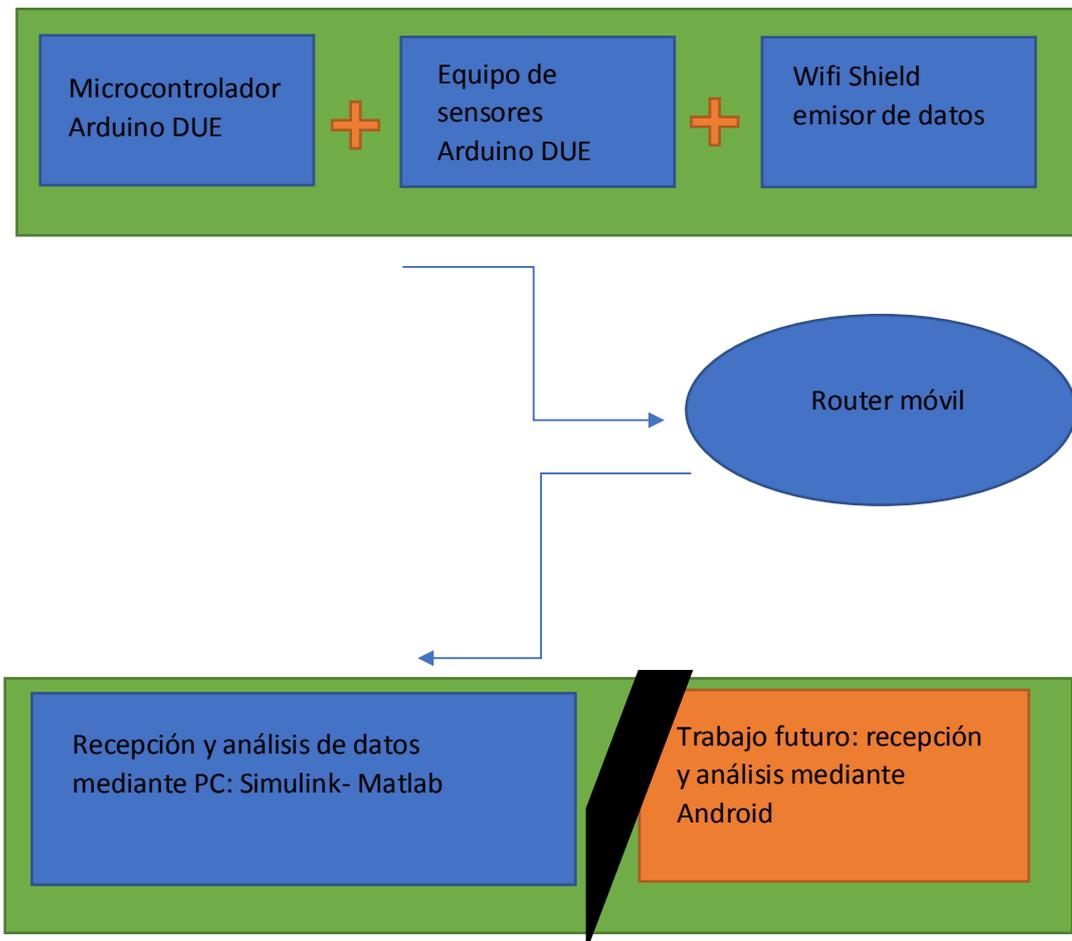
2. Estructura del prototipo

Ahora se va a detallar de forma general la estructura del prototipo de sensor de velocidad para el movimiento del press de banca.

En primer lugar, se tiene un microcontrolador que contiene el código para obtener los datos de interés durante el movimiento. Este microcontrolador extrae esos datos con ayuda de unos sensores que van, en este caso, unidos a la barra. Estos sensores son la parte fundamental del prototipo, ya que son los que captan movimiento mediante la aceleración lineal, los ángulos de desvío respecto a la posición inicial y además la profundidad de la repetición para ver si todas las repeticiones son iguales y no hay trampeo. Junto a lo mencionado se tiene también el módulo de transmisión de datos, en concreto WiFi, para poder enviar las variables de interés a otros dispositivos para poder procesar esos datos e interpretarlos de la manera adecuada. La red WiFi utilizada será una creada mediante un móvil smartphone Android.

En segundo lugar, se tiene el aparato receptor de los datos, en este caso, como es un prototipo y por falta de tiempo y medios se ha considerado como receptor un ordenador y un software de cálculo potente que permita realizar algunas operaciones en tiempo real, con un tiempo de muestreo de 50 ms. Entonces se recibirán los datos principales: aceleración en Z, ángulos de muñecas y sobre la horizontal y la profundidad de repetición o distancia mediante ultrasonidos.

En resumen, se tiene el equipo emisor: microcontrolador, equipo de sensores para captación de movimiento, posición y ángulos y el dispositivo WiFi para poder enviar los datos. De enlace se podría utilizar cualquier red WiFi, pero para hacerlo más seguro y personal se ha decidido poner un smartphone en modo router. Y por último el equipo receptor de datos para su procesamiento: un pc con software adecuado.



2.1 Equipo emisor

2.1.1 Microcontrolador Arduino Due

Una de las principales cuestiones del proyecto era en qué plataforma se iba a programar y qué ventajas y beneficios podrían obtenerse de ella y no de otra plataforma.

En primer lugar, la disponibilidad de placas Arduino Due por el departamento, estaban en “stock” y no haría falta comprar material extra, si hubiera surgido alguna urgencia.

La facilidad que nos permite el lenguaje C aplicado a Arduino mediante su programa IDE para escribir y cargar el código directamente a la placa Arduino es de gran ayuda, con librerías extensas y miles de aplicaciones rondando por la red; cualquier ingeniero electrónico ha coqueteado con este tipo de aparatos alguna vez durante su carrera y no está mal profundizar e intentar mejorar los códigos hechos en otras ocasiones.

El Arduino Due tiene una serie de características que lo hacen apropiado tales como:

- 54 E/S digitales, 6 de las cuales PWM
- 12 entradas analógicas
- 2 salidas analógicas (DAC)
- Poder operar a partir de 3.3V de alimentación
- 512 KB de memoria Flash
- 96 KB de memoria SRAM
- Frecuencia de reloj de 84MHz
- Muchas shield en el mercado acoplables



Ilustración 15 Arduino DUE

El precio, aunque no se tuvo que comprar a propósito, es bastante económico, y teniendo en cuenta la buena robustez, aunque el uso alguna vez ronde lo inadecuado, se puede decir que es buena opción.

Por lo tanto, se tiene una placa de dimensiones aceptables, para un prototipo inicial, con un lenguaje de programación conocido, con unas librerías accesibles para todo el mundo, con un software gratuito y ligero en velocidad computacional y con un precio razonablemente competitivo.

2.1.2 Bateria sensores

2.1.2.1 9 Axis Motion Shield

Esta placa es un sensor de orientación absoluto, de él se pueden obtener datos de ángulos, velocidad angular y aceleraciones angulares y lineales. Las características principales de este elemento son:

- Acelerómetro triaxial de 14 bits
- Giroscopio triaxial de 16 bits
- Sensor geomagnético triaxial
- Información de cuaterniones
- Ángulos de Euler
- Rotación Vectorial
- Aceleración lineal
- Gravedad vectorial
- 5 V de alimentación, 50 mW de consumo

Se ha utilizado en mayor medida para la aceleración lineal en el eje Z o vertical. De ahí, mediante simulink se ha integrado la aceleración y se ha conseguido la velocidad lineal en el eje vertical del movimiento. También se ha utilizado dos de los ángulos de Euler para indicar las desviaciones de la muñeca y barra respectivamente. Para la aceleración lineal hay unas funciones que se han obtenido de las distintas librerías donde había problemas ejemplo:

- `AcZ=mySensor.readLinearAccelZ()`. De aquí se obtiene la aceleración lineal en Z, ya cancelado el efecto de la gravedad.

Para los ángulos de rotación sobre las muñecas y el desbalance de los dos brazos se han utilizado los ángulos de Euler de las librerías:

- `Roll=mySensor.readEulerRoll`
- `Pitch=mySensor.readEulerPitch`



Ilustración 16: 9 Axis Motion Shield

2.1.2.2 Ultrasonidos HC SR04

Este sensor se ha puesto para ver si el rango de movimiento de cada repetición es el mismo, así evitar diferencias significativas en la profundidad de las repeticiones durante la serie; para ello se pondrá este sensor de distancia y se sacará una gráfica posteriormente en simulink.

Algunas de las características principales son:

- Ángulo de medición efectivo <math><15^\circ</math>
- Detección de 2cm-4m según fabricante
- Resolución de 3-30 mm
- Frecuencia de trabajo de 40 KHz

Este sensor irá mandando pulsos y rebotarán, se programa con el “echo” y el “trigger” y alimentándolo a 5 V. No interesa la medida absoluta, sino que en la gráfica saber si todas las repeticiones han sido a la misma profundidad.



Ilustración 17 Sensor de ultrasonidos

2.1.3 WiFi Shield

La aplicación práctica de la placa para el WiFi Shield es conseguir transmitir los datos de aceleración, ángulos y demás datos para el posterior procesamiento en el pc. Algunas características del Wifi Shiel son:

- Compatible con Arduino due
- Voltaje de 5 V
- Conexión vía 802.11b/g
- Encriptación WEPy WPA2
- Conexión mediante puerto SPI
- Posibilidad de memoria externa SD

Aquí se programa para que se conecte a la red que se ha diseñado y que la contraseña sea la correcta para que se conecte automáticamente al WiFi generado por el router, en este caso el móvil.

Como los datos que se van a recibir son de 8 bits, se van a mandar todos los datos en una escala de 0 a 255 de resolución, haciendo simples operaciones matemáticas se harán y se desharán este tipo de arreglos.



Ilustración 18: WiFi Shield

2.2 Equipo receptor: Matlab & Simulink

Matlab es un software muy potente, permite tratar innumerables operaciones y además tiene una programación bastante sencilla. En primer lugar, decir que la parte central del programa en Matlab se hace en el Simulink. Se utiliza el simulink principalmente para dos cosas: procesar la información que proporcionan los sensores para obtener información útil; y visualizar esta información en una interfaz de usuario fácil de entender.

La primera parte del programa en simulink es donde se reciben los datos enviados vía WiFi desde el Arduino y recibidos por un bloque que los trata y los separa en 4 mediante un demultiplexor. A partir de ahí se hacen una serie de operaciones:

- Sacar la distancia
- Meter la aceleración en un subsistema bastante complejo para poder hacer operaciones y sacar la velocidad, es decir, la integral de la aceleración
- Acondicionar los ángulos Roll y pitch para su interpretación posterior

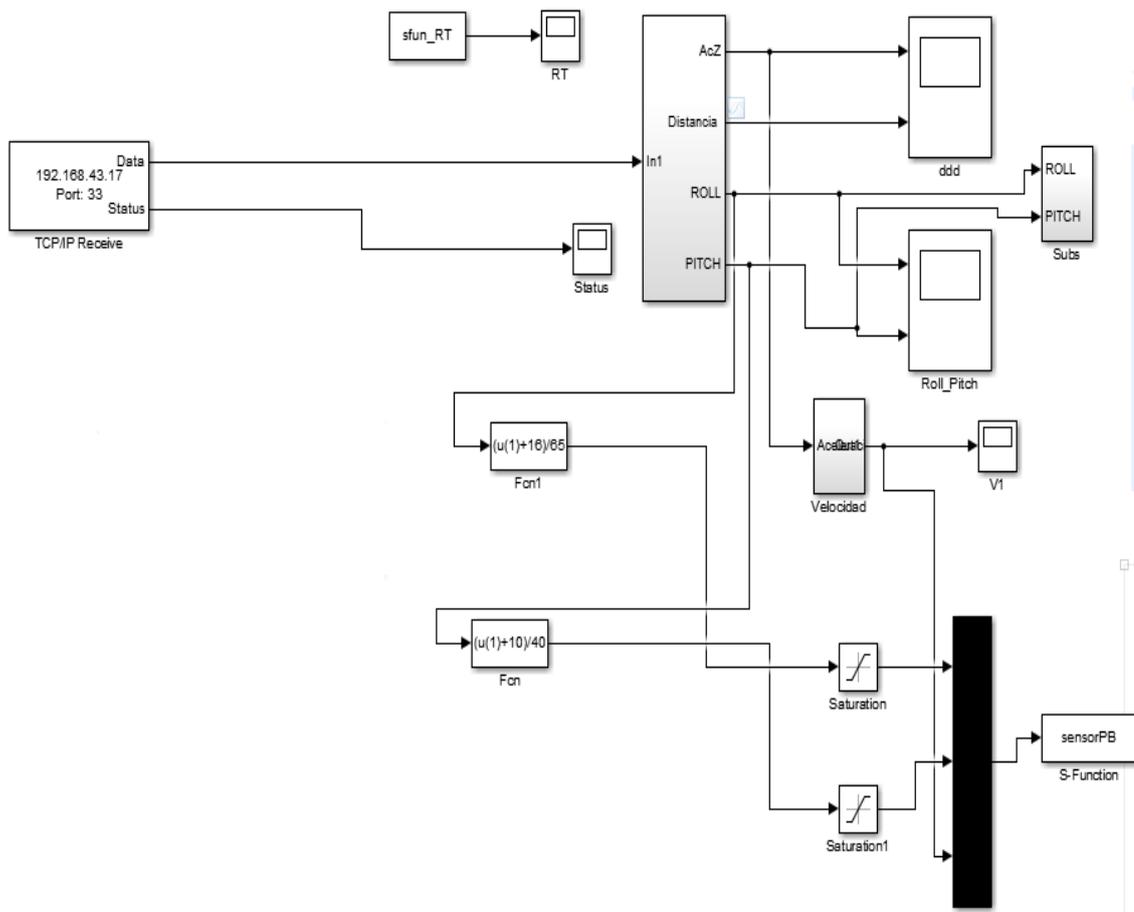


Ilustración 19: Página principal del diagrama simulink

Dentro del subsistema velocidad que aparece en la imagen anterior se tiene el siguiente diagrama de bloques donde se hacen una serie de operaciones lógicas, se filtran, se integra la

aceleración para obtener, finalmente, la velocidad respecto al tiempo. Es una trama bastante compleja y se ha tenido que hacer de esta manera para evitar el llamado error acumulativo por ruido que estropea la gráfica por el ruido inevitable de los sensores. En este caso se resetean los bloques cuando suceden eventos como la velocidad negativa durante cierto tiempo o no hay movimiento en la aceleración (con lo cual se fija a 0 en vez de una rampa descendente por culpa del ruido). Así, se consigue obtener la velocidad de la manera más limpia y efectiva posible que nos permite simulink, son soluciones mediante ensayo-error que parecen coherentes con el resultado final. Hay que decir que hay veces donde se descontrola un poco el sensor y sale alguna cosa extraña o inesperada pero la idea general que proporciona la velocidad siempre es clara.

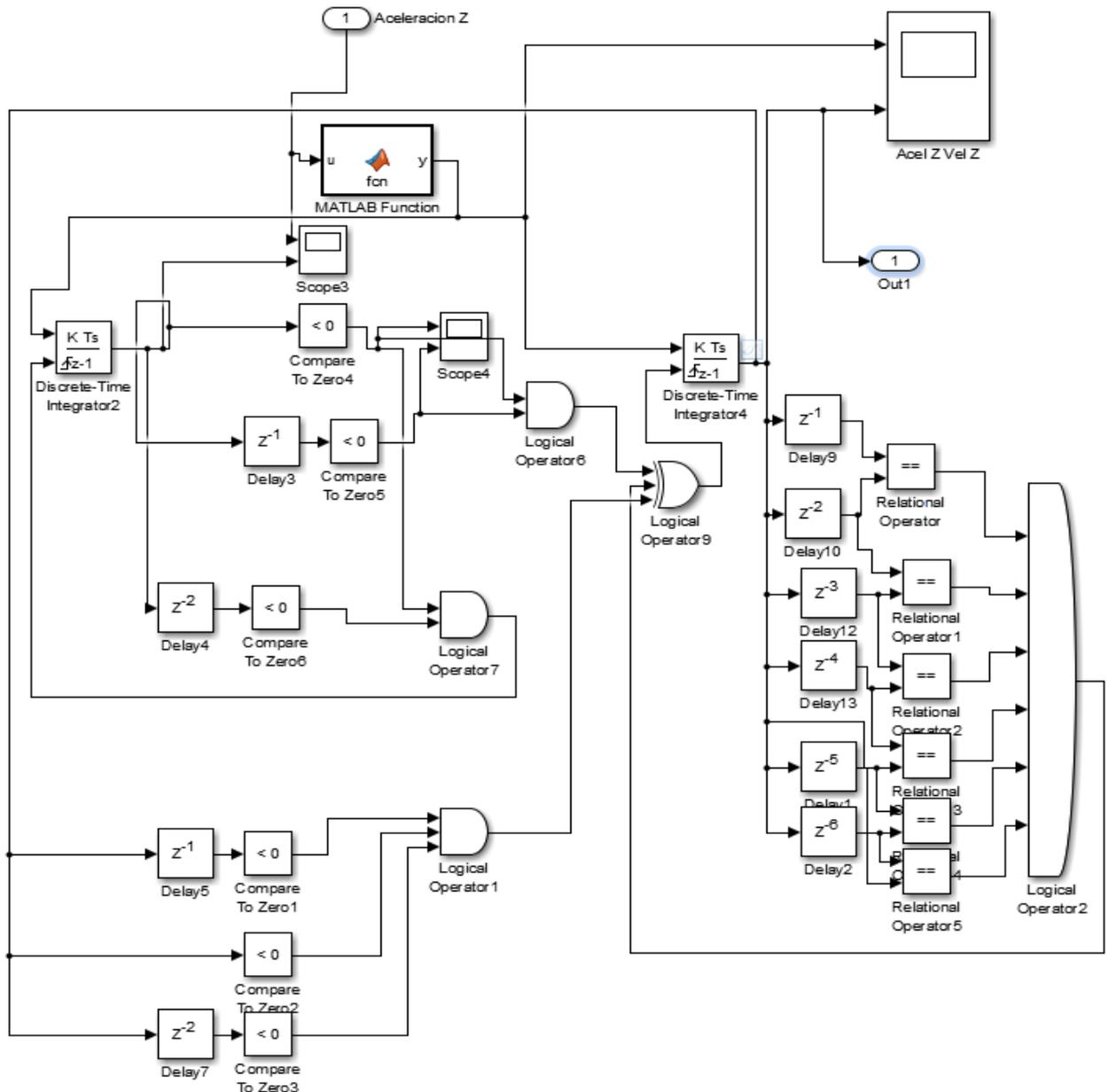


Ilustración 20: Subsistema para la obtención de la velocidad vertical a partir de la aceleración

Como se ve la aceleración entra en una función de Matlab donde se filtra para que no esté vibrando, es decir, se le limita una zona muerta para que la señal sea estable en 0 a valores muy bajos. Después sencillamente se va a dos integradores donde se comparan distintos

periodos de muestreo de ambos y si son iguales hace que se resetee el integrador; esto se hace para evitar el error prolongado en forma de pendiente debido al ruido que se genera en la señal. Así, se consigue tener una señal bastante fiable y que si no se mueve nada esté en 0 y no en constante pendiente negativa. Grosso modo es una trama de operaciones lógicas con and y xor que permiten resetear dichos integradores para que la gráfica de la velocidad, la que sale del integrador numero 4 (el de la derecha) sea lo más fiel al movimiento real y que la gráfica quede bien presentada.

Una vez hecho lo anteriormente comentado, se procede a enviar los datos que sean de interés a una función que lo guardará en Matlab y a partir de ahí se puede crear una interfaz donde sea más visual y menos complejo para el usuario interpretar qué está pasando en tiempo real, como también se observa en la utilización de una función que permite tener todo en tiempo de muestreo de 50 ms.

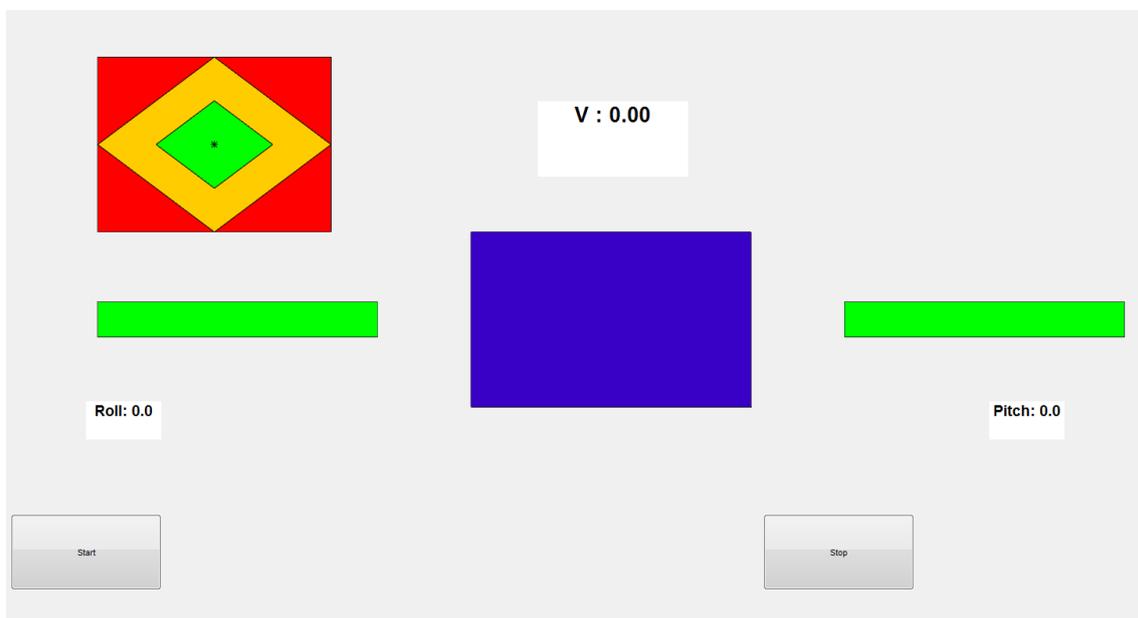


Ilustración 21: Interfaz principal

Las barras laterales tienen un color verde por defecto, se van hacia los verdes chillones si los ángulos son positivos, por el contrario, hacia los rojos si por los ángulos son los inversos. Una cosa igual pasa con el rectángulo del centro que varía en función de la velocidad. Además, el botón de empezar y parar también está un display arriba la velocidad instantánea.

Esto es algo muy básico y visual, la gente de corriente puede ver interpretar que si las barras no son muy verdes ni muy rojas durante un tiempo el movimiento está bien hecho en cuanto a posicionamiento angular. También si las velocidades no son muy bajas mediante el código de colores y el display es que ha sido un movimiento efectivo. Pero para los que deseen más información, a parte de la interfaz se ha decidido implementar unas operaciones y sacar unas tablas de resultados. Para ello se saltará una imagen donde se deberá introducir el peso y al darle a stop se ejecutará el archivo que contiene las tablas de resultados. En ellas se podrá ver:

- La evolución de la velocidad respecto al tiempo
- La evolución de los ángulos respecto al tiempo
- La profundidad de repetición respecto al tiempo

Además, en el workspace se sacarán los valores de pico de la velocidad, aceleración, fuerza máxima desarrollada y potencia.

2.3 Diseño del soporte mediante NX

Gracias a la tecnología de diseño en 3D se ha podido diseñar las piezas necesarias para el acople del sensor con su alimentación incluida con la barra del ejercicio. Se hicieron varios prototipos para ver si encajaban correctamente antes de mandar a realizar la definitiva. El soporte está formado en 4 partes fundamentales. Las dos partes que rodean la barra primero se imprimieron con un grosor de 1 cm para ver si encajaban a la perfección, después se realizaría la impresión definitiva con la impresora del instituto.

Ahora se mostrarán dichas piezas en su modelo 3D y en su anexo se adjuntarán los planos de las mismas.

Parte del acople del sensor ultrasónicos en 3D:

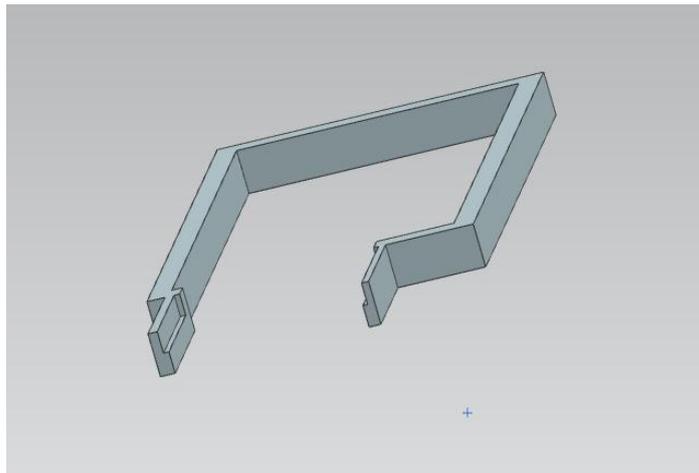


Ilustración 22: Pieza superior y soporte del sensor ultrasónicos

Pieza 2 o bloque central:

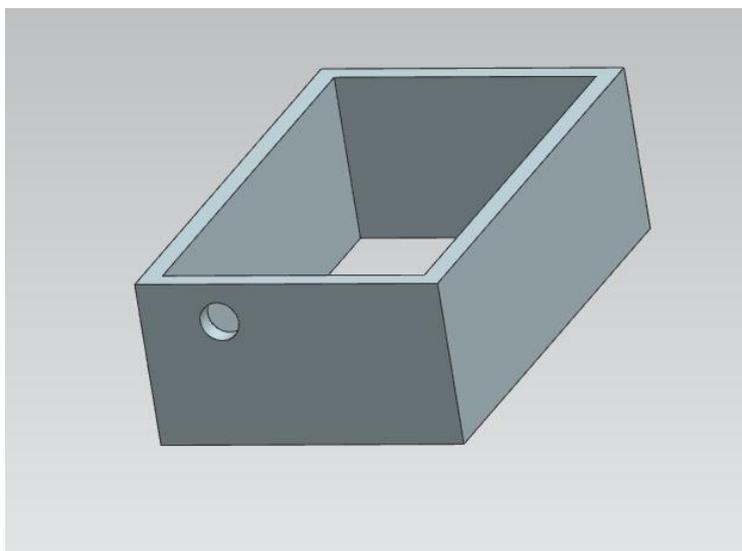


Ilustración 23: Pieza central que protege el sensor

Pieza 3 o bloque de acople por debajo:

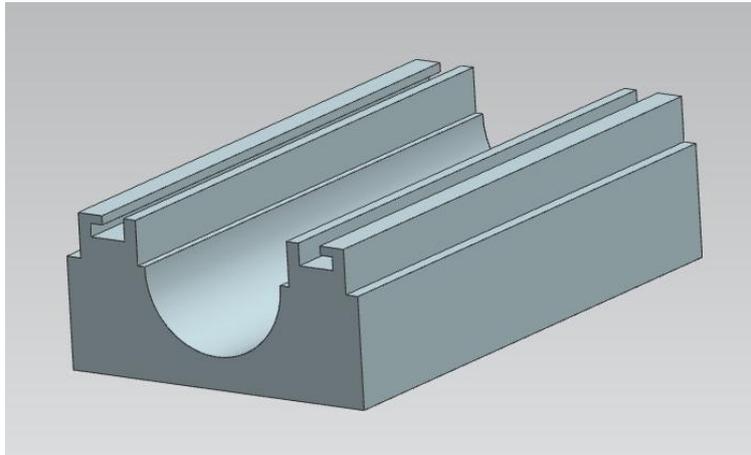


Ilustración 24: Pieza que queda por debajo de la barra de press de banca

Pieza 4 o acople por arriba:

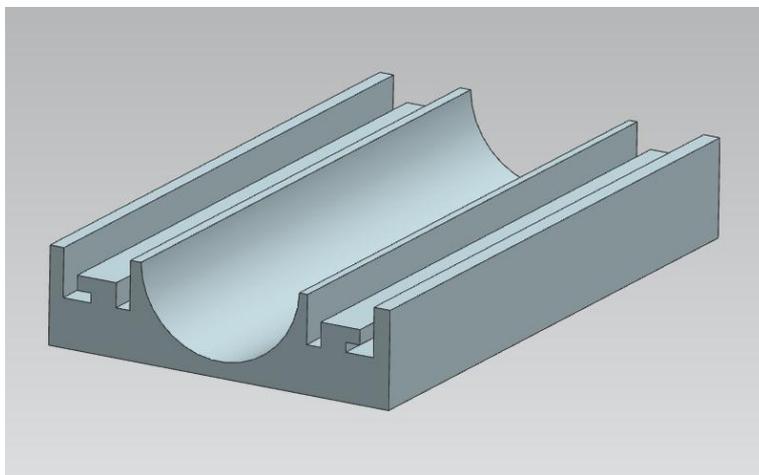
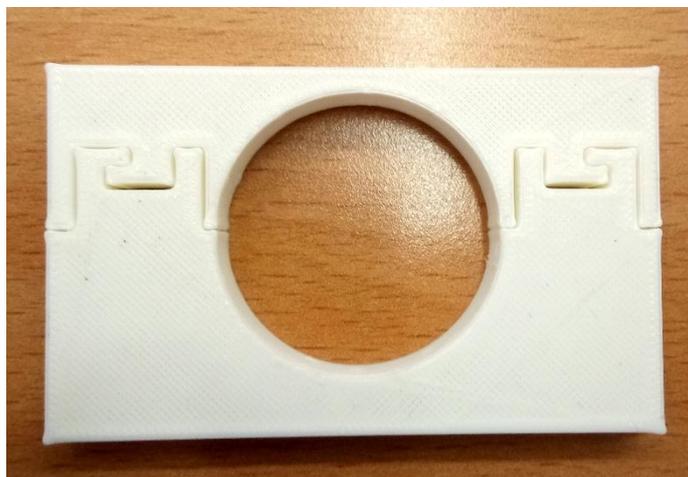
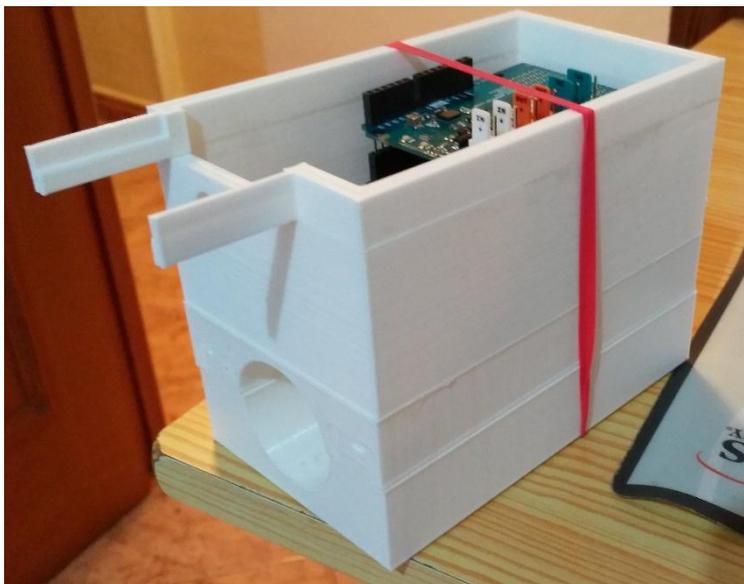


Ilustración 25: Pieza que va encolada con la pieza central y se acopla con la pieza anterior rodeando la barra

Primera impresión de las piezas 1 y 2 para ver si encajaban:

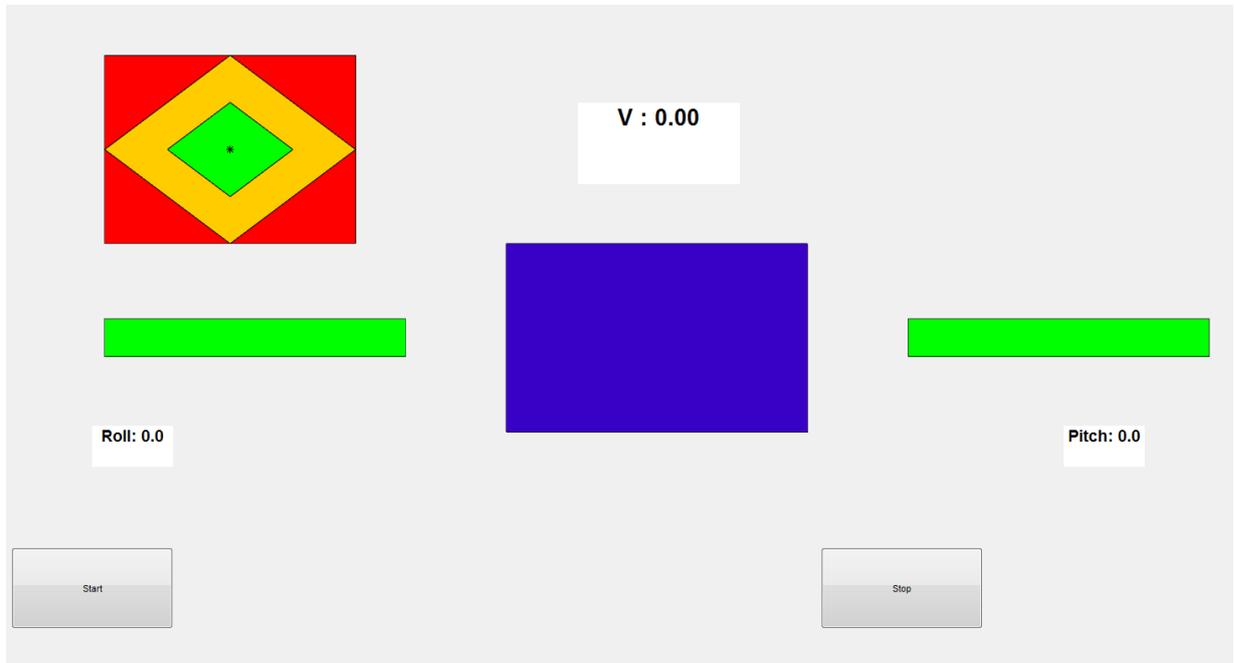


Piezas montadas en la versión definitiva:



3. Resultados Experimentales

En primer lugar, se desea volver a mostrar la interfaz realizada para el seguimiento de la ejecución en tiempo real:



Es una interfaz simple donde se tienen 3 displays para la visualización de la velocidad instantánea (centro), desviación sobre las muñecas (izquierda) y desviación sobre la horizontal (derecha). Además 4 figuras que indicarán de la velocidad (centro), los laterales un código de colores para los ángulos apoyando la ilustración de arriba en la izquierda donde el asterisco mostrará la posición angular en tiempo real. Todo esto además de los botones Start/Pause y Stop.

3.1 Ejecución incorrecta por inclinación lateral

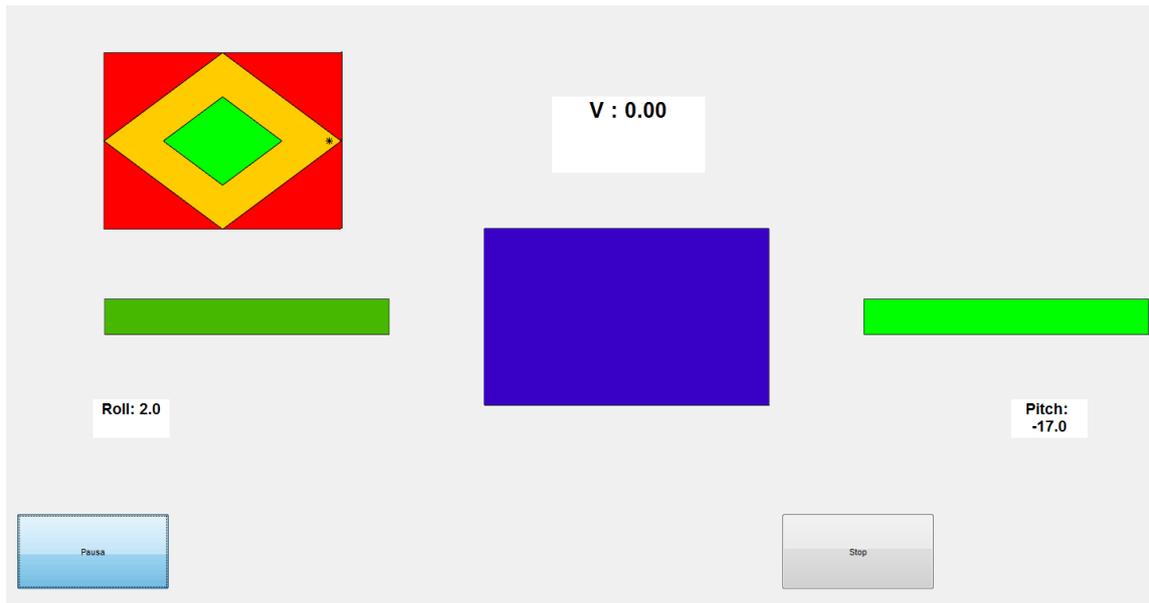


Ilustración 26: Ángulo sobre la horizontal demasiado grande en un sentido

Como se puede observar en el color de la barra de la derecha, verde muy vivo y su ángulo sobre la horizontal, en ese momento -17° y además en el dibujo de arriba a la izquierda se ve como se desplaza el asterisco que nos indica la situación angular del sensor.

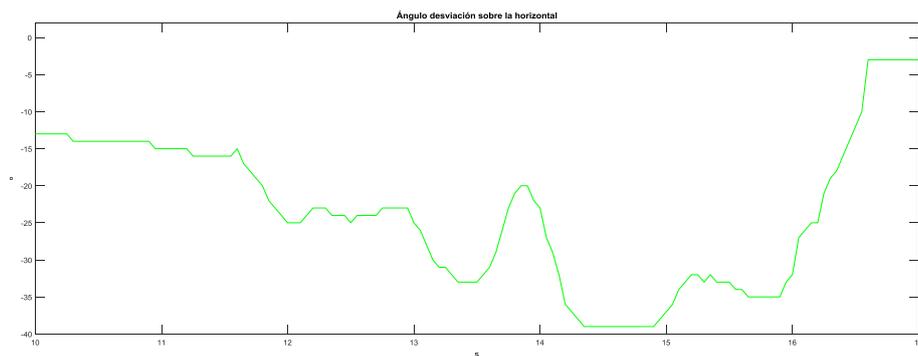


Ilustración 27: Evolución del ángulo de la horizontal respecto al tiempo

En la figura de abajo se ve la evolución de este ángulo respecto al tiempo de manera forzada. Se puede ver que el ángulo máximo en este caso ronda los 39° con signo negativo.

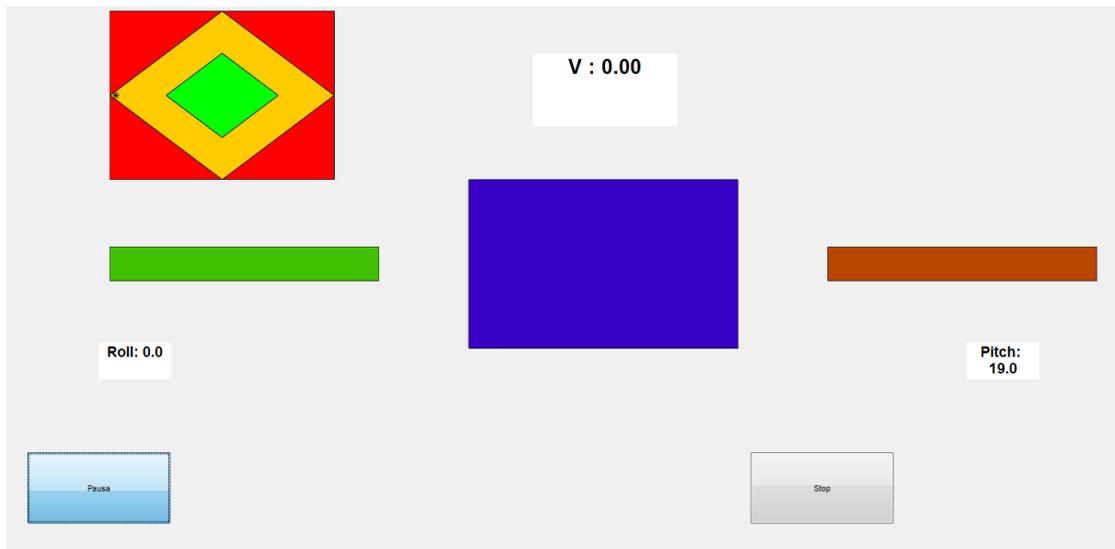


Ilustración 28: Ángulo peligroso en la horizontal

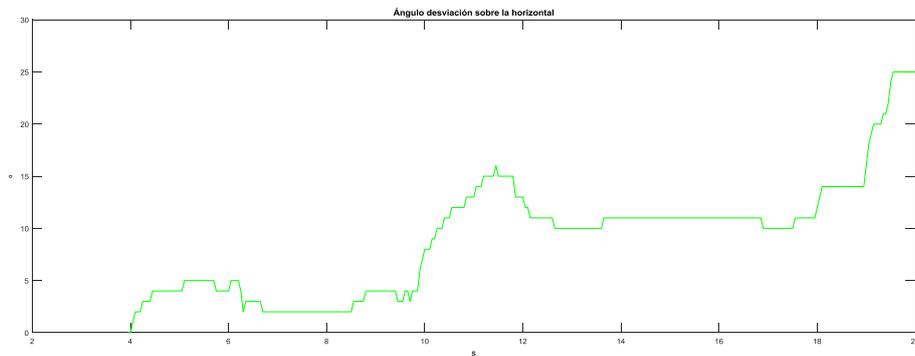


Ilustración 29: Evolución del ángulo respecto al tiempo

En estas dos figuras se ve lo mismo, pero en sentido contrario a lo anteriormente comentado. El ángulo es peligroso, lesivo y, por tanto, la ejecución es mala. En este caso el brazo que antes subía con más velocidad aquí es el brazo rezagado.

A continuación, se mostrará una imagen donde el ángulo es mínimo y la ejecución no llega a ser peligrosamente mala. Como vemos en la imagen siguiente, la barra está a medio camino entre el rojo y el verde, con lo cual la interfaz nos indica que el ángulo no es algo muy llamativo, es cierto que este prototipo es bastante sensible y hay momentos donde puede haber picos: en este caso de 12°. Bastante menos de la mitad de los grados límite de antes. Por lo cual, la ejecución puede ser correcta, la evolución del ángulo nos indica unos pocos grados de media, aproximadamente 6°.

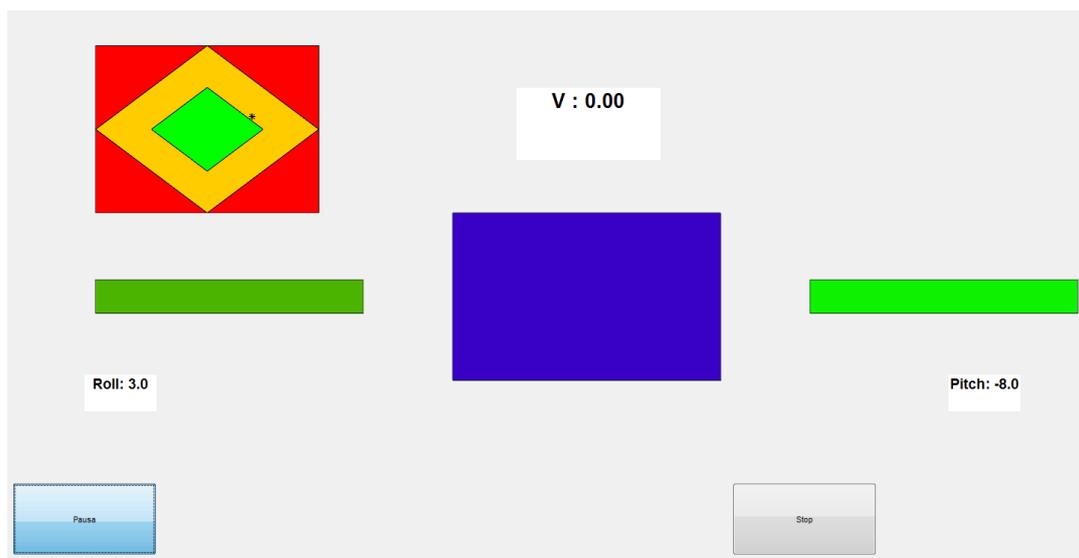


Ilustración 30: Ángulo bastante bueno sobre la horizontal

Aquí se debe poner el foco en lo que marca el pitch -8.0 y también en el asterisco donde se aprecia que no está centrado, pero que sobre se sitúa la marca verde en la horizontal. Es decir, si no se tuviera desviación sobre las muñecas (roll) el asterisco pisa zona verde, zona hábil y correcta.

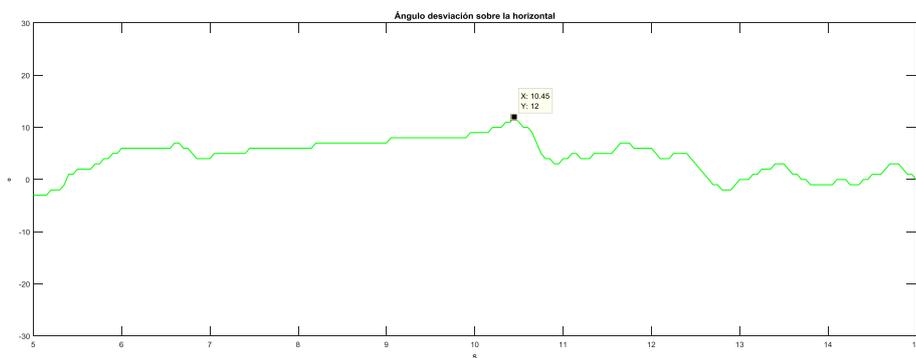


Ilustración 31: Evolución del ángulo sobre la horizontal

3.2 Ejecución incorrecta por inclinación frontal

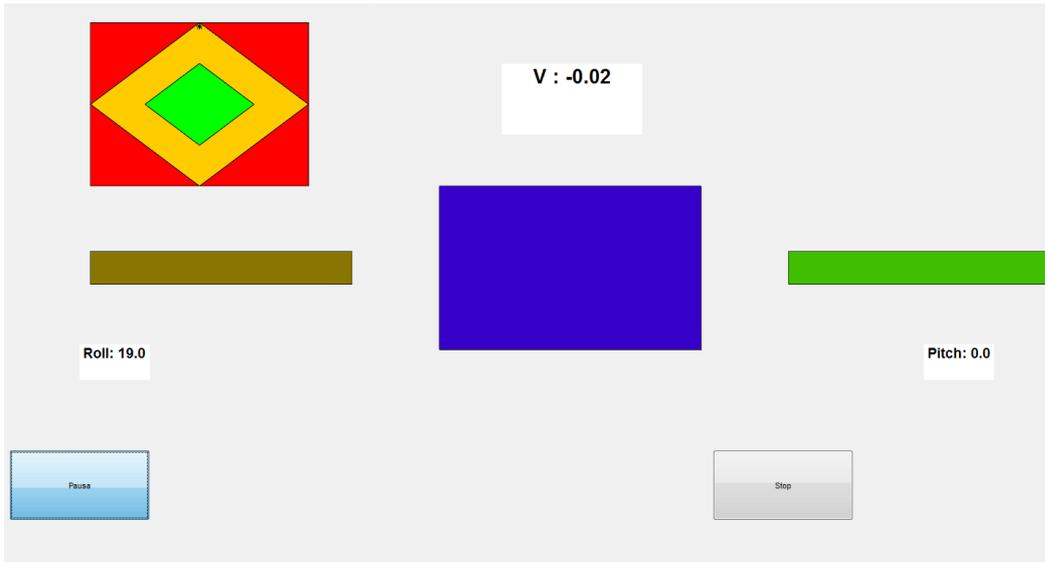


Ilustración 28: Desviación leve sobre las muñecas

Como se aprecia en la imagen la barra está de un ocre, esto nos indica que la desviación de la barra respecto a la muñeca es notable pero puede ser aún mayor, como se mencionó anteriormente, los colores extremos son el rojo vivo y el verde fuerte, en las siguientes imágenes se verá una desviación muy marcada, peligrosa en un sentido y en otro. El asterisco está en el límite entre una mala ejecución y una muy mala ejecución.

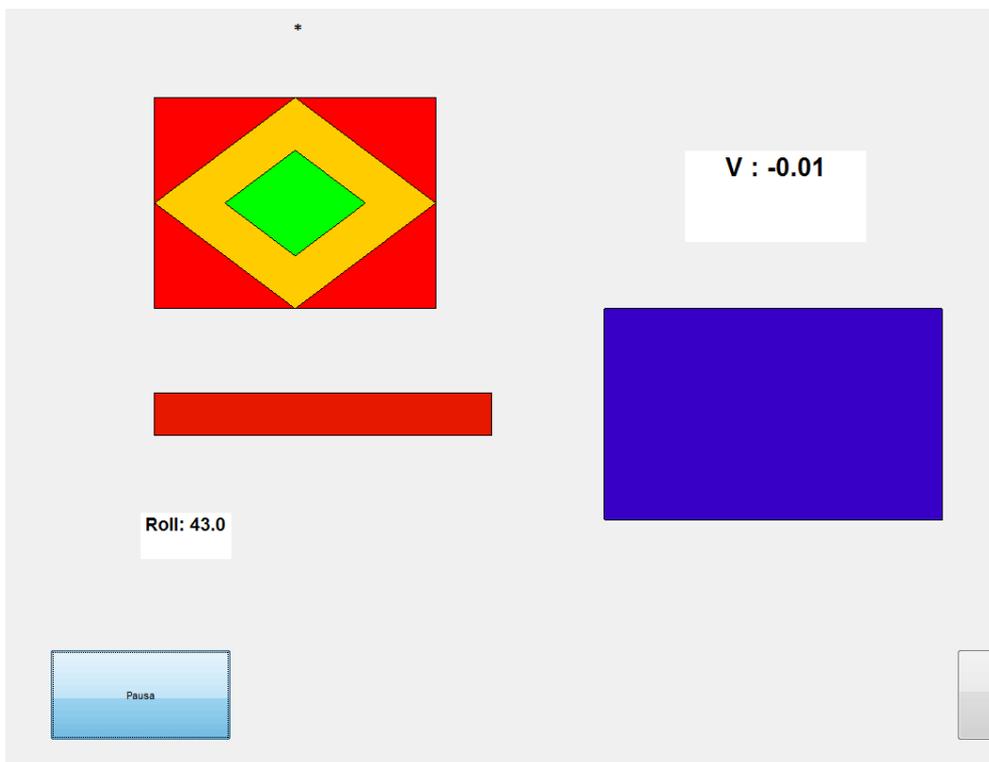


Ilustración 32: Desviación peligrosa en un sentido sobre las muñecas

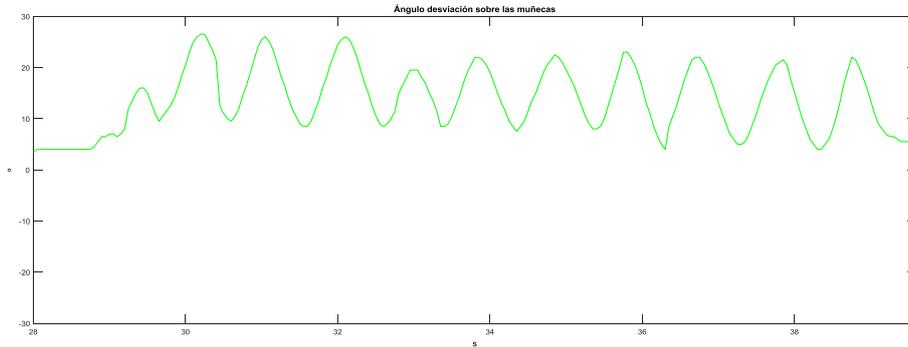


Ilustración 33: Evolución del ángulo respecto a la muñeca

Como se ha visto, en esta ocasión el rojo es muy marcado, el ángulo es peligroso y lesivo para las articulaciones. Además, el asterisco se sale por completo de la ilustración por lo cual se da a entender que el movimiento está realizado mal. Debajo se muestra la evolución de varios giros hacia ese mismo sentido. Se ve que el ángulo máximo diseñado, en las gráficas, está sobre los 27º, casi un 1/6 de giro de circunferencia completa.

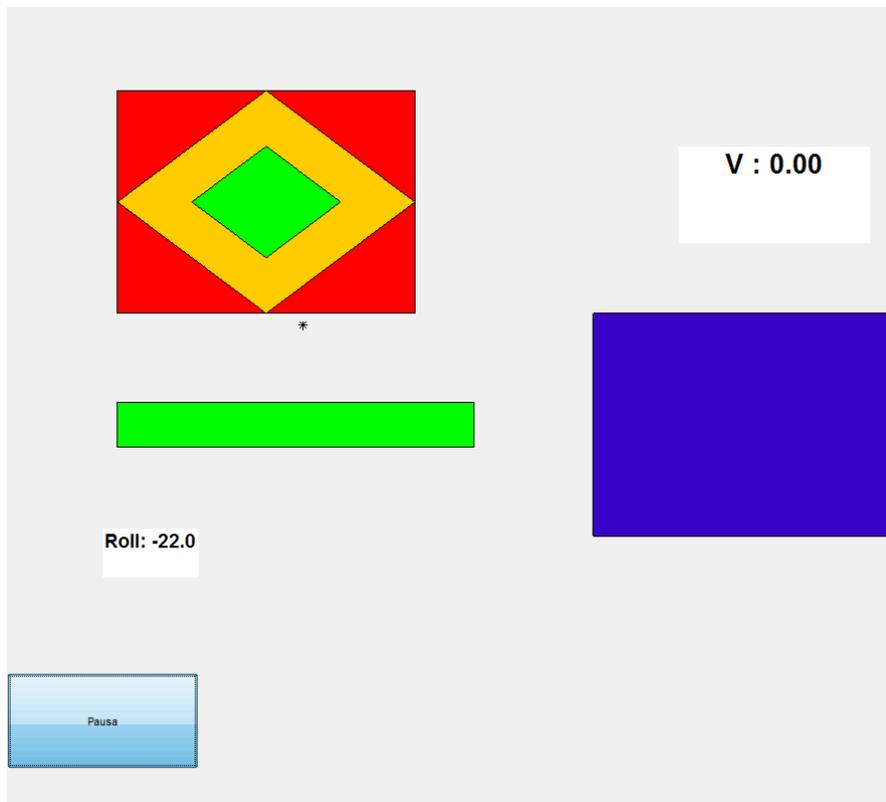


Ilustración 34: Ángulo límite en el sentido contrario

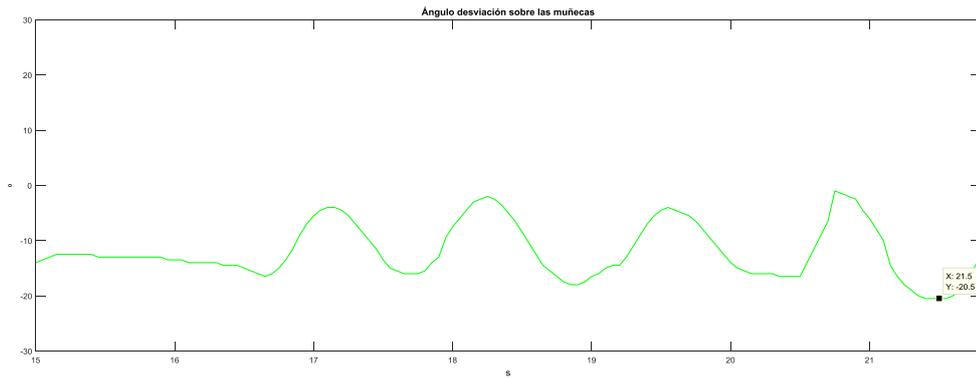


Ilustración 35: Evolución del ángulo de la muñeca respecto al tiempo de ejecución

En estas dos imágenes, se contraponen el giro del anterior experimento, el sentido es el contrario, el verde se ve claramente que el ángulo es notablemente peligroso +20º pero en negativo.

3.3 Ejecución incorrecta por pérdida de velocidad

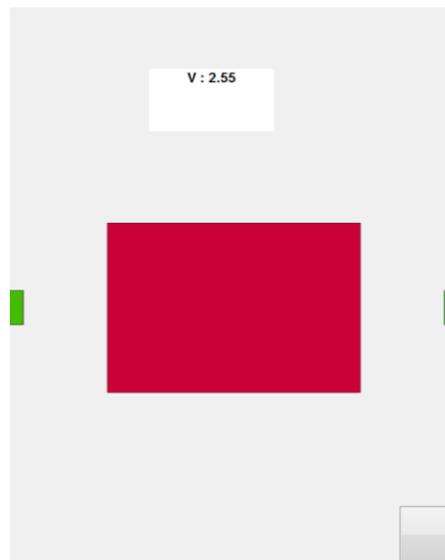


Ilustración 36: Indicador de velocidad en la interfaz

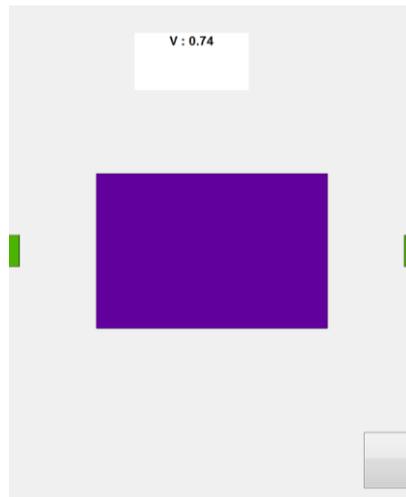


Ilustración 37: Velocidad menor que la anterior

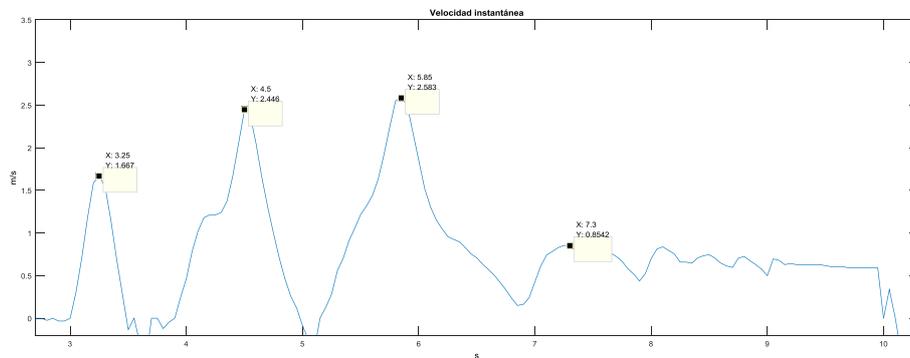


Ilustración 38: Evolución de la velocidad a través del tiempo

En esta situación vamos a interpretar los resultados de la gráfica. En primer lugar, se ve una repetición bastante fluida con una velocidad pico de más de 2.25 m/s, la segunda es todavía más rápida: 2.45 m/s y la tercera es el punto álgido de la serie donde se alcanza la máxima velocidad de toda la ejecución con más de 2.55m/s. Lo que pasa luego es un intento de otra repetición o pero que se acaba fallando y la barra no llegará a su punto alto para confirmar la cuarta repetición.

Se ha conseguido capturar la pantalla en el momento de la tercera repetición y velocidad máxima durante toda la serie, la interfaz nos indica con el tono de rojo fuerte que la velocidad estaba siendo alta y la ejecución hasta el momento perfecta.

La imagen de la captura siguiente de la interfaz, la segunda imagen es una velocidad es una velocidad ya menor en comparación a las anteriores.

Lo malo ha sido el desenlace de la serie. Se sabe que se ha intentado la cuarta repetición, pero no se ha conseguido. Todo esto es una dramatización de lo que puede ocurrir en un entreno pesado.

3.4 Ejecución correcta vs incorrecta por profundidad de repetición

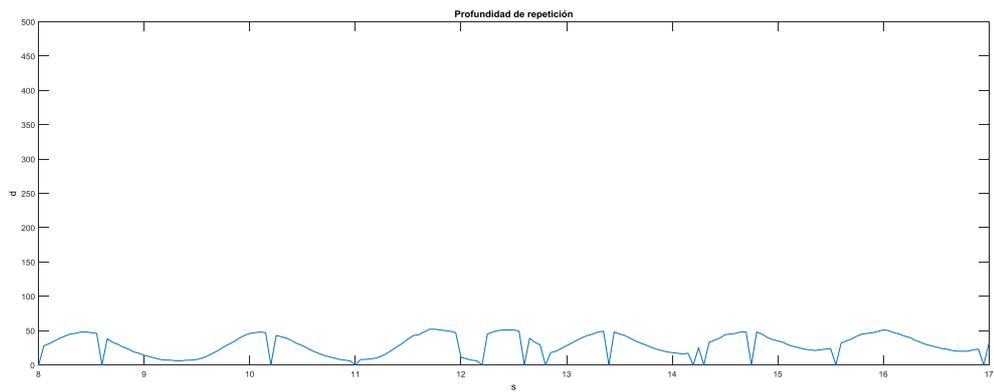


Ilustración 39: Evolución del sensor de ultrasonidos (1)

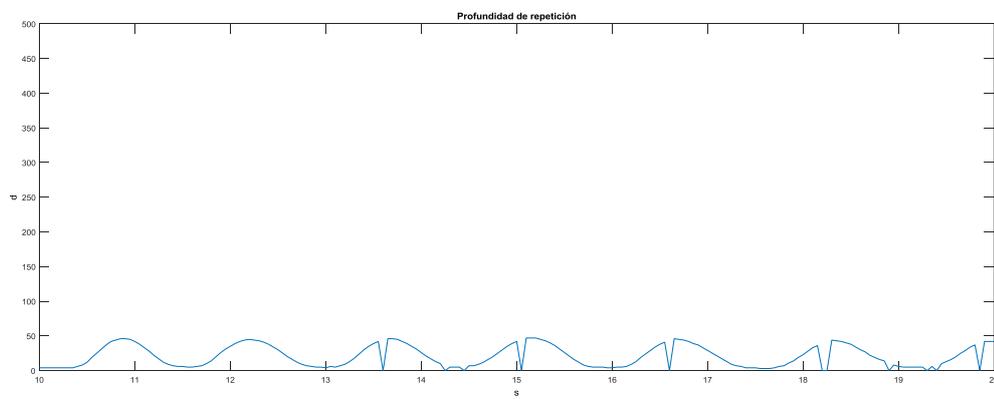


Ilustración 40: Evolución del sensor de ultrasonidos (2)

En primer lugar, en este apartado se va a evaluar la profundidad de repetición que influye directamente en el rango de movimiento (ROM). Es decir, si la profundidad de repetición siempre es la misma, con un margen de error mínimo, se considerará la ROM completa. Ahora se procede a comparar las dos imágenes. Se ve claramente que la imagen 2 es bastante más constante en todos los sentidos, picos máximos, valles mínimos, frecuencia...etc. Lo principal es que el máximo y el mínimo sean siempre constante en la medida de lo posible, cosa que en la primera imagen hay momentos que no lo parece. Si se tuviera que decir cuál ha sido mejor serie en cuanto al rango de movimiento sería claramente la segunda prueba.

3.5 Ejecución correcta

Como se ha detallado anteriormente, una buena ejecución debe tener una velocidad aceptable durante el mayor rato posible, es decir, durante el máximo número de repeticiones del levantamiento. Se estima que por debajo del 15/20 % de pérdida de velocidad se debería cortar la serie para evitar la fatiga en exceso.

Por otro lado, los ángulos han de ser lo menor que sea posible, es cierto que puede haber un pequeño margen donde los ángulos bailen ya que el movimiento vertical 100% no es, se parece más a una J ligeramente invertida. Aquí se estimará correcto un ángulo menor de 12° en valor absoluto.

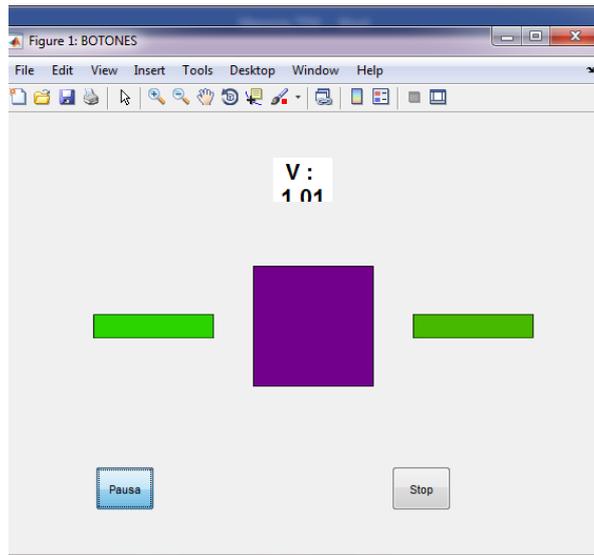


Ilustración 41: Interfaz en miniatura durante una ejecución

Ejemplo 1

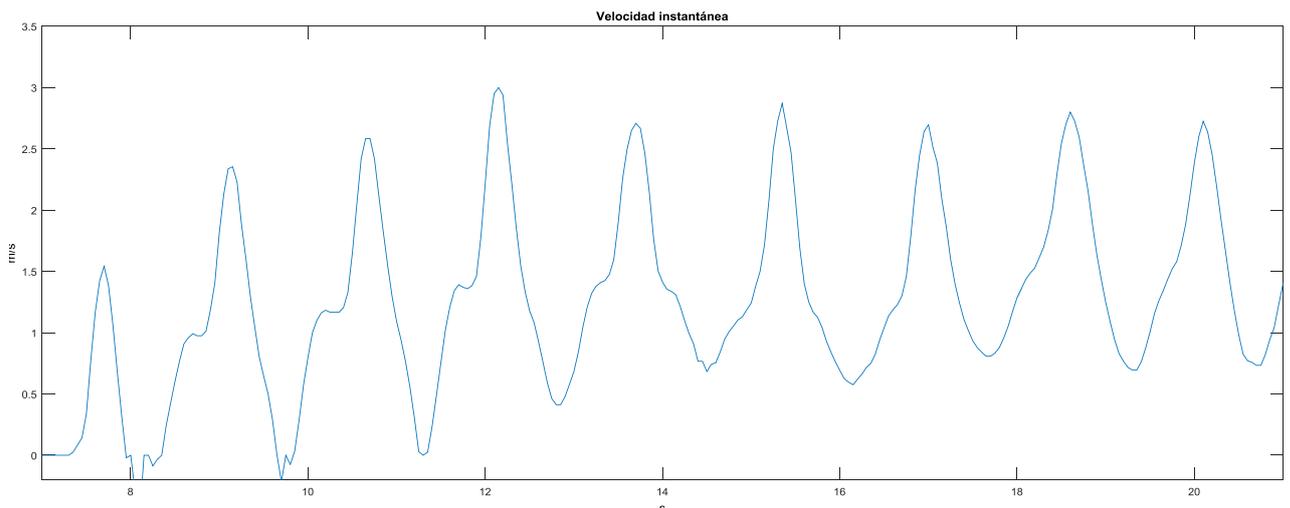


Ilustración 42: Evolución de la velocidad a través del tiempo

Como se puede ver aquí, el atleta realiza 9 repeticiones, la máxima velocidad se consigue en la tercera repetición con valor de 3 m/s. Las siguientes series se mantienen en una gran velocidad en torno a 2.7-2.8 m/s, bastante alejados del 2.4 m/s que podría indicar fatiga.

Ejemplo 2

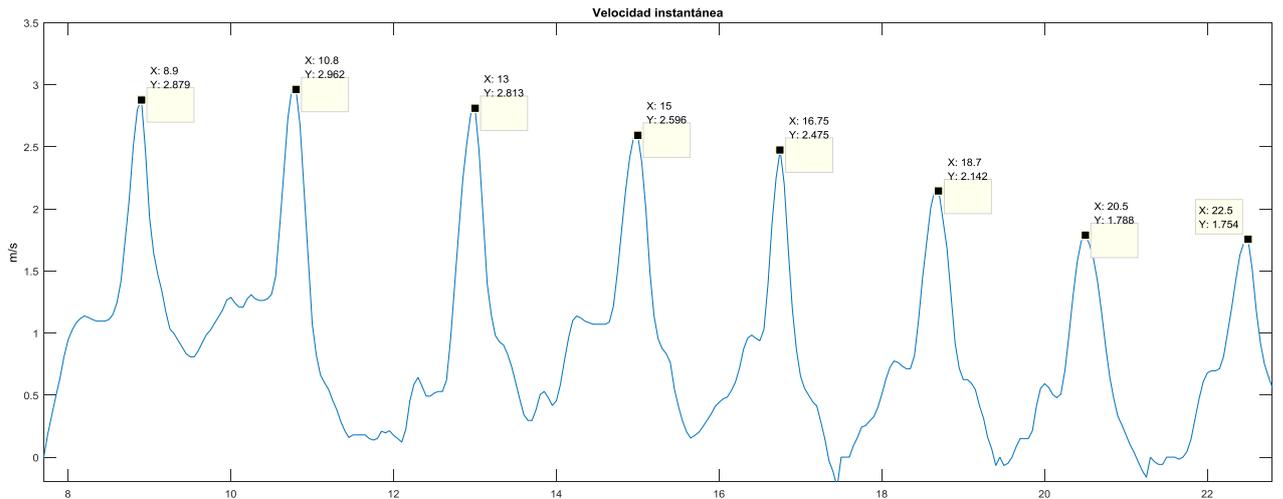


Ilustración 43: Evolución de la velocidad respecto al tiempo

En este caso, el sujeto realiza 8 repeticiones donde se ve que la fatiga hace mella y cada vez va perdiendo algo de velocidad, siendo las 3 primeras muy veloces rondando los 3 m/s y las dos últimas menores de 1.8 m/s como se puede ver en la imagen. Como datos de interés de esta serie se tiene:

- Velocidad máxima: 2.975 m/s
- Aceleración máxima (medida directamente del sensor): 10.25 m/s²
- Fuerza máxima desarrollada (suponiendo levantamiento con 100 kg): 1025 N
- Potencia máxima desarrollada: 297.5 W

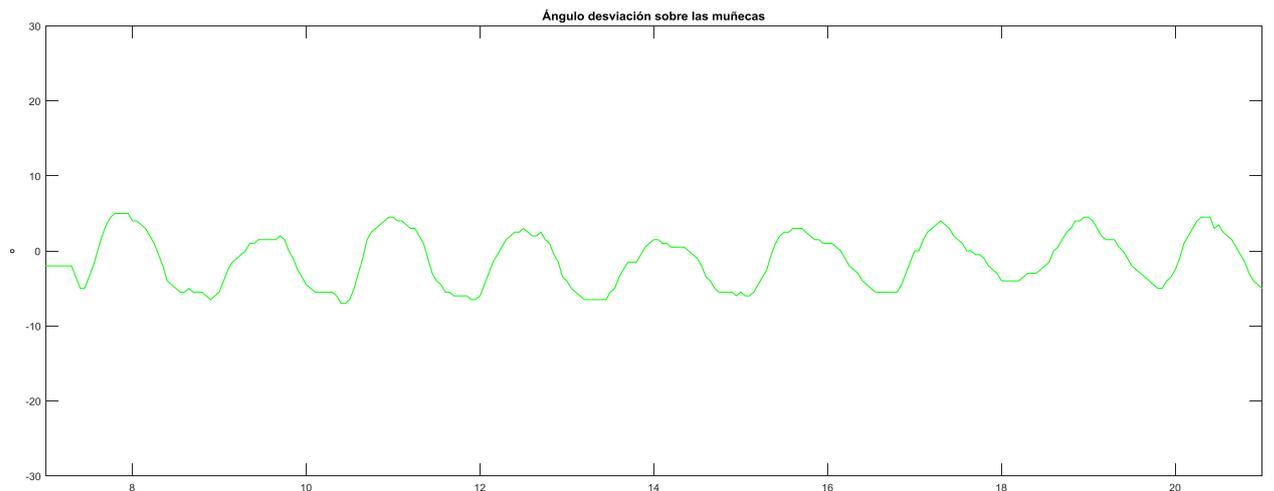


Ilustración 44: Evolución del ángulo en las muñecas

Como se ha comentado antes, este movimiento sería perfectamente válido siempre por debajo del 8º en valor absoluto.

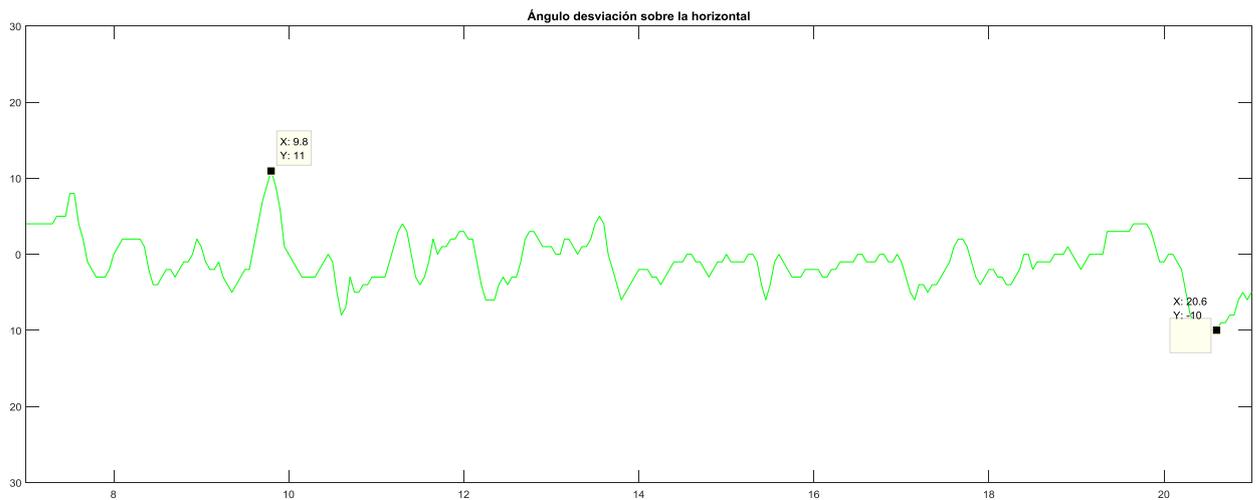


Ilustración 45: Evolución del ángulo de la horizontal

Si se adoptaba lo anterior, que por debajo de 12° era un buen movimiento, aquí lo tenemos, donde se está cerca en alguna ocasión; se podría concluir en que es un movimiento válido, correcto, para nada peligroso o lesivo, pero mejorable.

3.6 Imágenes de una prueba en tiempo real



Ilustración 46: Movimiento de press de banca en el punto más bajo



Ilustración 47: Movimiento de press de banca en punto inicial y más alto



Ilustración 48: Sujeto entrenando press de banca con parada isométrica

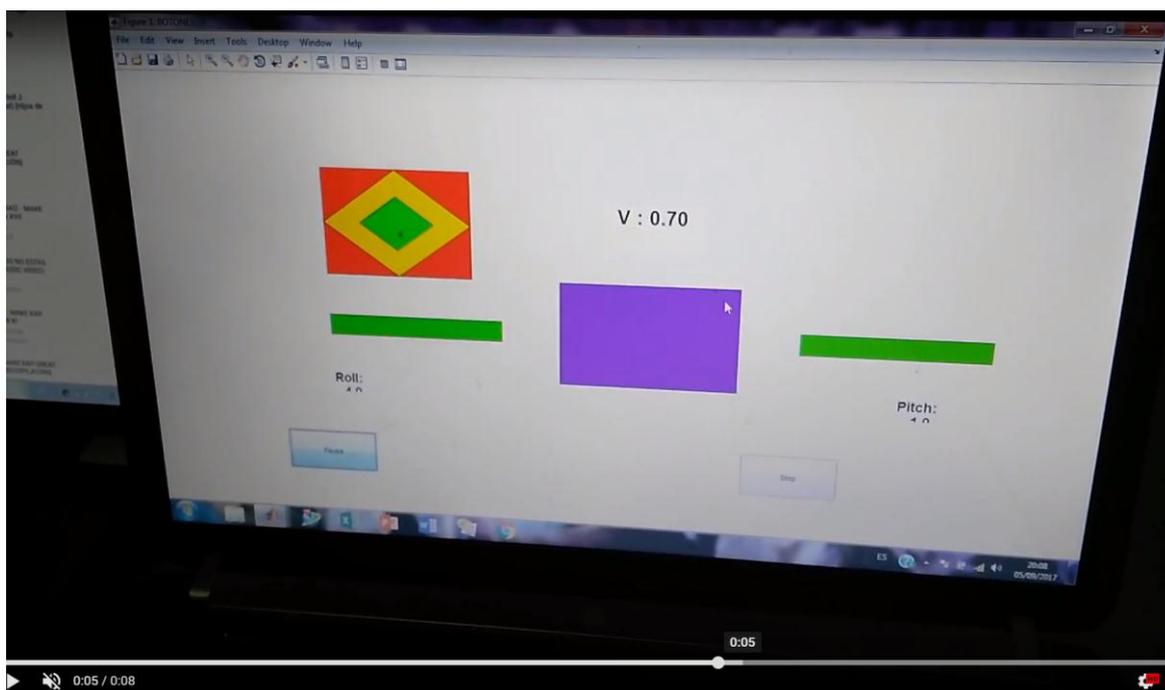


Ilustración 49: Imagen extraída de un vídeo mientras el sujeto anterior realizaba la prueba

4. Conclusiones

4.1 Evaluación de los resultados y del global del proyecto

Para empezar cabe destacar que los elementos utilizados son parte de la llamada ingeniería de bajo coste o coste reducido. Debido a ello, se nota sobre todo en el tamaño. Que es la principal pega y desventaja el proyecto. No tanto el peso, ya que puede rondar el 0.5% del peso levantado para competición de un atleta medio en una categoría media y en la rehabilitación el peso no es importante si no la ejecución, la cadencia e intentar mejorar poco a poco la zona dañada.

En cuanto a los resultados, son buenos en la medida de lo posible, los tiempos de muestreo adecuados de sobra y la resolución es suficiente para este tipo de aplicación, ya que es posible mejorarla, pero con este nivel se cubre muy bien las necesidades del atleta de competición y se sobrepasa lo requerido para una rehabilitación normal en un hombro o brazo.

4.2 Mejoras posibles al proyecto en un futuro

En primer lugar, como ya se ha comentado, decir que el proyecto es sólo una primera versión de algo que podría llegar a ser muy sofisticado y útil, pero por falta de tiempo y de recursos no se ahonda en dicho objetivo. Por lo cual, se está mejorando últimamente en el campo de la rehabilitación de accidentes donde las personas pierden movilidad o después de operaciones los ligamentos y tendones se debilitan y no están al mismo nivel la estructura musculoesquelética que el tejido conectivo. Debido a ello se pueden encontrar deficiencias y asimetrías a la hora de realizar según que ejercicios o puede ocurrir que se necesite un seguimiento más profundo y profesional de cómo se desarrollan los movimientos para rehabilitar a aquellas personas que lo necesiten. Para la mejora del proyecto inicial se van a enumerar las ideas que el diseñador tiene en mente para las futuras mejoras:

- En primer lugar, el paso del proyecto a Android para mejorar la facilidad de manejo y no tener que tener siempre un pc con Matlab para su monitorización.
- Después se pueden aumentar el número de sensores para tener un mejor seguimiento de las diferentes trayectorias en ejercicios multiarticulares y así ver los momentos exactos de debilidad, fallo, cualquier asimetría, punto de estancamiento etc...
- La reducción de todos los componentes del proyecto en cuanto a tamaño, estudiar una medida más sofisticada y potente pero que, a su vez, sea de menor tamaño para poder sacarle el máximo rendimiento al sensor sin molestar lo más mínimo posible. Para ello se podría estudiar la idea de integrarlo todo en una placa.
- Entrando en las funcionalidades nuevas se podría incluir una unidad emisor-receptor para que monitorice el trayecto en 3 dimensiones, así se puede evaluar el movimiento en el espacio sin necesidad de analizarlo en cada plano.
- También podría ser útil el añadir alarmas de carácter sonoro para avisar de ciertas cosas al ejecutor, inicio, mal, final, sonido agradable para trayectoria o despliegue de fuerzas bueno.

En definitiva, la idea es conseguir minimizar el número de lesiones en los gimnasios, mejorar el rendimiento deportivo de los atletas que lo deseen en sus respectivas competiciones y, si no es posible evitar la lesión ayudar a rehabilitar de una forma más efectiva y menos dependiente del profesional, para no tener que desplazarse y ocupar material sanitario público...



Ilustración 50: Profesional evaluando la ejecución de un ejercicio de tracción vertical Jalón tras nuca

5.Presupuesto

Presupuesto Detallado del Prototipo			
Parte 1: Parte emisora			
Ref	Cantidad	Descripción	Precio €
Mat1	1	Microcontrolador arduino Due	36
Mat2	1	9 Axis Motion Shield para Arduino	40
Mat3	1	WiFi Shield para Arduino	38
Mat4	1	Sensor ultrasonidas HCSR04	6
Parte 2: Receptor y procesado de datos			
Mat5	1	HP Pavilion DVD6 i7, 8 GB RAM, 256 GB SSD	800
Mat6	1	Licencia Matlab 2015a	200
Mat7	1	Smartphone utilizado como router WiFi	85
Parte 3: Impresión 3 D de los soportes del sensor			
Mat8	1	4 piezas por fabricación aditiva	75
Trabajo de ingeniería			
Mat9	120h	120 h de trabajo a 10€/h	1200
Importe final			
			2480
Importe final Con IVA + 18% beneficio para el creador			
			3447,2

Cabe destacar que el presupuesto en una empresa de ingeniería con licencias de matlab y el equipo correcto podría reducir el coste de manera considerable.

6. Pliego de Condiciones

Objeto

El objeto de este documento es el de fijar las condiciones técnicas mínimas que ha de tener el proyecto de prototipo de sensor inalámbrico de velocidad de press de banca mediante microcontrolador Arduino. Contando también la interfaz gráfica de computadora que acompaña para la interpretación de los datos.

El documento aborda el trabajo con los dispositivos electrónicos utilizados y sus programas creados para el funcionamiento correcto del sistema. También aparecerán algunos aspectos relacionados con el montaje del sensor.

Especificaciones de materiales y equipos

Placa Arduino Arduino DUE

Descripción

La placa Arduino DUE es la utilizada para el programar el sensor con un microcontrolador ATmega328P, alimentación entre los 7 y 12 V, 54 entradas/salidas digitales, memoria de almacenamiento de 512 KB, memoria SRAM de 96KB y 84MHz de velocidad de reloj. Se recomienda la utilización de dicho modelo, aunque se pueden utilizar placas de mayor rendimiento que la descrita, de la familia de Arduino (nunca de menor rendimiento).

Control de calidad

Se conectará al ordenador y se comprobará que el IDE de Arduino reconoce a la placa. Observaremos el puerto virtual COM que se genera, para su utilización en un futuro.

9 Axis Motion Shield

Descripción

Esta es la Shield que proporcionará información sobre la orientación y movimiento de la barra. Debe contener: acelerómetro triaxial de 14 bits, giroscopio triaxial de 16 bits, sensor geomagnético triaxial, información de cuaterniones y ángulos de Euler; información sobre la rotación vectorial. Aceleración lineal y gravedad vectorial. Debe estar alimentado a 5V.

Control de calidad

Se conectará al ordenador y se acoplará con la Arduino due, deberán comprobarse los leds para ver si está todo bien conectado, para su utilización en un futuro.

WiFi Shield

Descripción

Esta es la Shield que permitirá transmitir los datos de manera instantánea mediante tecnología WiFi. Deberá contener conexión vía 802.11b/g , encriptación WEPy WPA2 y posibilidad de conexión mediante puerto SPI.

Control de calidad

Se conectará al ordenador y se acoplará con la Arduino due, deberán comprobarse los leds para ver si está todo bien conectado, para su utilización en un futuro.

Sensor HCSR04

Descripción

Esta es el sensor de distancia ultrasónico compatible para tecnología Arduino. En lugar de este sensor se podría utilizar otro con un rango de trabajo de 10 mm a 300 mm y con una resolución menor a los 4mm.

Control de calidad

Se conectarán correctamente los pines del sensor a la placa WiFi, en la parte superior de todo, y se correrá el programa demo_distancia para comprobar el correcto funcionamiento del sensor.

Móvil router

Descripción

Aquí sirve cualquier terminal móvil que sea capaz de generar una red Wifi vía 802.11b/g , encriptación WEPy WPA2 y posibilidad de conexión mediante puerto SPI.

Control de calidad

Se conectará el PC y la Arduino mediante el programa WiFi para ver si es capaz de transmitir los datos al móvil y este al PC.

Ordenador

Descripción

El equipo principal consta de un ordenador con procesador Intel core i7 de 4 núcleos a 2 GHz, 8 GB de memoria RAM DDR3 y 1 GB de gráficos en DDR5. A su vez se necesita el programa Matlab by Mathworks con Simulink y todas las librerías. También es necesario que el programa esté instalado en un SSD con al menos 15 GB libres para operaciones. Valdría cualquier pc con características superiores a las mencionadas.

Control de calidad

Se encenderá el ordenador, se cargará el fichero de simulink y se conectará a la red Netgear generada por el router móvil.

Piezas por fabricación aditiva

Descripción

Se han diseñado expresamente 4 piezas para el acople del sensor y su soporte en la barra. El diámetro está diseñado para las barras oficiales de competición de 34mm. Las medidas pueden ser reducidas en algunos aspectos mientras no pierdan robustez.

Control de calidad

Se montarán todas las piezas. 3 de las cuales van encoladas formando una unión inseparable, la otra encaja a la perfección con la que rodea la barra para que no se despegue de la misma el prototipo.

Ejecución y control de calidad

En primer lugar, hay que observar que ninguna de los componentes tenga ningún desperfecto a nivel visual; seguidamente y, después de haber comprobado uno por uno tal y como se ha detallado en el apartado anterior en el control de calidad, se procederá a unir las piezas del soporte si no se había realizado previamente. En siguiente lugar, se deberá montar conectar la Arduino Due, ya conectada la 9 axis motion shield y la wifi shield , a la batería externa y colocarla dentro de las piezas que se han unido. A continuación, colocar los pines de alimentación y masa al sensor de ultrasonidos, y conectar el “echo” al pin 6 y “trigger” al pin 7 de la placa Shield Wifi. Finalmente se coloca este equipo sobre la barra y la pieza nº 4 se coloca por abajo.

La segunda fase de la ejecución concierne a la parte receptora. El PC se deberá encender y cerrar todos los programas que sean posibles para solo tener abierto el Matlab. Se conectará a la red WiFi correspondiente al router móvil. Se cargará el fichero de simulink y se ejecutará. Saltará la interfaz por pantalla. Cuando se de a inicio, el sujeto ya puede descender la barra para iniciar las repeticiones que sean convenientes, al finalizar el ejercicio se dará a Stop.

Pruebas de Servicio

Como es bastante complicado evaluar por partes el funcionamiento de las distintas partes, se decide poner las pruebas de servicio como un simulacro de repetición real, es decir, sin los soportes, sin barra, sin atleta. Simplemente ejecutando el programa y moviendo el equipo sensor con los brazos hacia distintos lados y ver que la gráfica da algo coherente, comprobando a su vez, que la interfaz muestra bien los detalles a seguir.

7 Bibliografía

1. <https://www.strengthandconditioningresearch.com/2014/07/08/injury-strength-sports/>
2. <http://www.garywu.net/9-unexpected-benefits-from-weightlifting/>
3. <https://simplifaster.com/articles/olympic-lifts-importance-peak-velocity-recommended-guidelines/>
4. <http://suppversity.blogspot.com.es/2017/02/bench-press-study-higher-weight-less-of.html>
5. <https://www.strengthandconditioningresearch.com/exercises/bench-press/>
6. https://openi.nlm.nih.gov/detailedresult.php?img=PMC4633267_jhk-47-51f2&req=4
7. <https://powerexplosive.com/importancia-de-la-parada-en-la-fiabilidad-de-las-mediciones-isoinerciales/>
8. <https://powerexplosive.com/velocidad-de-ejecucion/>
9. 1. González-Badillo, J.J., Ribas J. (2002). Bases de la programación del entrenamiento de fuerza . Barcelona: Editorial Inde
10. 2. Chulvi, I. (2010). Muscular failure training in conditioning neuromuscular programs. J.Hum. Sport Exerc:5(2):196-213.
11. 3. Sánchez-Medina, L. González-Badillo, J.J. (2010). Importance of the propulsive phase in strength assessment. Int J Sports Med. 31:123-129
12. 4. Balsalobre, C., & Jiménez-Reyes, P. (n.d.). Entrenamiento de fuerza: Nuevas perspectivas y metodología. iBooks.
13. <https://kinetic.com.au/stories/gymaware/mladen-jovanovic-estimating-1rm.html>

Anexo I Código Arduino IDE

```
#include "NAxisMotion.h" //Contains the bridge code between the
API and the Arduino Environment
#include <Wire.h> //
#include <SPI.h>
#include <WiFi.h> //wifi shield
#include <NewPing.h> //ultrasonidos
#define TRIGGER_PIN 7 // Arduino pin tied to trigger pin on the
ultrasonic sensor.
#define ECHO_PIN 6 // Arduino pin tied to echo pin on the
ultrasonic sensor.
#define MAX_DISTANCE 300 // Maximum distance we want to ping for (in
centimeters). Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup
of pins and maximum distance. //no wifi
NAxisMotion mySensor; //Object that for the sensor //no
wifi
unsigned long lastStreamTime = 0; //To store the last streamed
time stamp //no wifi
const int streamPeriod = 40; //To stream at 25Hz without
using additional timers (time period(ms) =1000/frequency(Hz)) //no wifi
bool updateSensorData = true; //Flag to update the sensor
data. Default is true to perform the first read before the first
stream //no wifi

char ssid[] = "NetGear"; // your network SSID (name)
char pass[] = "abcd1234"; // your network password
int status = WL_IDLE_STATUS; // the Wifi radio's status
WiFiServer server(33);

int t1 = 0 ;
int t2 = 0;
int T = 50; // Periodo de
muestreo (milisegundos)
unsigned long distancia;

byte AcZ ;

int EuP ;

void setup()
{
    Serial.begin(115200); //Initialize the Serial Port to view
information on the Serial Monitor
    I2C.begin(); //Initialize I2C communication to
the let the library communicate with the sensor.

    mySensor.initSensor(); //The I2C Address can be changed
here inside this function in the library
    mySensor.setOperationMode(OPERATION_MODE_NDOF); //Can be
configured to other operation modes as desired
```

```

    mySensor.setUpdateMode(MANUAL); //The default is AUTO. Changing
to manual requires calling the relevant update functions prior to
calling the read functions
//Setting to MANUAL requires lesser reads to the sensor
mySensor.updateAccelConfig();
updateSensorData = true;
Serial.println();
Serial.println("Default accelerometer configuration settings...");
Serial.print("Range: ");
Serial.println(mySensor.readAccelRange());
Serial.print("Bandwidth: ");
Serial.println(mySensor.readAccelBandwidth());
Serial.print("Power Mode: ");
Serial.println(mySensor.readAccelPowerMode());
Serial.println("Streaming in ..."); //Countdown
Serial.print("3...");
delay(1000); //Wait for a second
Serial.print("2...");
delay(1000); //Wait for a second
Serial.println("1...");
delay(1000); //Wait for a second

if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    while (true);
}
String fv = WiFi.firmwareVersion();
if ( fv != "1.1.0" )
    Serial.println("Please upgrade the firmware");
while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    status = WiFi.begin(ssid, pass);
    delay(10000);
}
server.begin();
Serial.print("You're connected to the network");
printCurrentNet();
printWifiData();
}

void loop()
{
    distancia = sonar.ping_cm();
    if (updateSensorData) //Keep the updating of data as a separate
task
    {
        // mySensor.updateAccel(); //Update the Accelerometer data
        mySensor.updateLinearAccel(); //Update the Linear Acceleration
data
        mySensor.updateEuler();
        //mySensor.updateGyro(); //Update the Gravity Acceleration data
        // mySensor.updateCalibStatus(); //Update the Calibration Status
        updateSensorData = false;
    }
    if ((millis() - lastStreamTime) >= streamPeriod)
    {
        lastStreamTime = millis();
    }
}

```

```

//AcX=((mySensor.readLinearAccelX()*12))+128 ;
//AcY=((mySensor.readLinearAccelY()*12))+128 ;
AcZ=((mySensor.readLinearAccelZ()*12))+128 ;

// EuH=((mySensor.readEulerHeading()*10))+100 ;
// EuR=((mySensor.readEulerRoll()*10))+100 ;
// EuP=((mySensor.readEulerPitch()*10))+100 ;

// Serial.print(" aX: ");
// Serial.print(((int)AcX-128)/12.0);
// Serial.print(" aY: ");
// Serial.print(((int)AcY-128)/12.0);
Serial.print(" aZ: ");
Serial.println(((int)AcZ-128)/12.0);

//server.write(AcX);
//server.write(AcY);
server.write(AcZ);

/* Serial.print(" EuH: ");
Serial.print(((int)EuH-100)/10.0);
Serial.print(" EuR: ");
Serial.print(((int)EuP-100)/10.0);
Serial.print(" EuP: ");
Serial.println(((int)EuR-100)/10.0);
*/
// Serial.print(" H: ");
// Serial.print(mySensor.readEulerHeading()); //Heading data 0-360
// Serial.print("deg ");

Serial.print(" R: ");
Serial.print(mySensor.readEulerRoll()); //Roll data -90 a +90
Serial.print("deg");

Serial.print(" P: ");
Serial.print(mySensor.readEulerPitch()); //Pitch data -180 a +180
Serial.print("deg ");

Serial.print(" d: ");
Serial.print(distancia); //Heading data 0-360
Serial.print("cm ");

server.write(distancia);
server.write((mySensor.readEulerRoll()+128));
server.write((mySensor.readEulerPitch()+128));

updateSensorData = true;

t2=millis() ; //
Temporización
if (t2-t1>T) {
    digitalWrite(13,HIGH) ; Serial.println(t2-t1) ;}
else {digitalWrite(13,LOW) ;}
while (t2-t1<T) {t2=millis() ;}
t1=millis() ;

}
}

```

```

void printWifiData() {
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);
  Serial.println(ip);
  byte mac[6];
  WiFi.macAddress(mac);
  Serial.print("MAC address: ");
  Serial.print(mac[5], HEX);
  Serial.print(":");
  Serial.print(mac[4], HEX);
  Serial.print(":");
  Serial.print(mac[3], HEX);
  Serial.print(":");
  Serial.print(mac[2], HEX);
  Serial.print(":");
  Serial.print(mac[1], HEX);
  Serial.print(":");
  Serial.println(mac[0], HEX);
}

void printCurrentNet() {
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());
  byte bssid[6];
  WiFi.BSSID(bssid);
  Serial.print("BSSID: ");
  Serial.print(bssid[5], HEX);
  Serial.print(":");
  Serial.print(bssid[4], HEX);
  Serial.print(":");
  Serial.print(bssid[3], HEX);
  Serial.print(":");
  Serial.print(bssid[2], HEX);
  Serial.print(":");
  Serial.print(bssid[1], HEX);
  Serial.print(":");
  Serial.println(bssid[0], HEX);
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.println(rssi);
  byte encryption = WiFi.encryptionType();
  Serial.print("Encryption Type:");
  Serial.println(encryption, HEX);
  Serial.println();
}

```

Anexo II Código Matlab archivos y diagramas de bloques

sensorPB.m (SFunction)

```

function [sys,x0]=sensorPB(t,x,u,flag)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Animación sensor Press Banca.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global roll pich cuad vel txt ff asterix locity rolln pitchn
% Variables globales
switch flag,

    case 2,
% Se ejecuta en cada paso de simulación

        r=u(1); p=u(2); v=u(3); rg=u(4); pg=u(5);
% Entradas de la animación

        if (t==0) set_param('Septiembre','SimulationCommand','Pause') ; end
;

        set(txt,'String',sprintf('t = %2.2f',t))

        kil=0;
% Modificación de propiedades

        % roll=fill([-2 -2 10 10],[-2 2 2 -2],[r (1-r) 0.5]);
        % pich=fill([16 16 28 28],[-2 2 2 -2],[p (1-p) 0]);
%capsula

set(roll,'Facecolor',[r (1-r) 0]);
set(pich,'Facecolor',[p (1-p) 0]);
set(locity,'Facecolor',[((v+1)/4.5) 0 (1-((v+1)/4.5))]);
set(vel,'String',sprintf('V : %1.2f',v));
set(rolln,'String',sprintf('Roll: %1.1f',rg));
set(pitchn,'String',sprintf('Pitch: %1.1f',pg));
set(asterix,'XData',[-10+(-pg/4)],'YData',[20+(rg/2)]);%-----
-----
drawnow ;
% ;; Dibuja ahora !!
    sys=[] ;

    case 0,
% Se ejecuta al iniciar la simulación

        colordef white;
% Creación pantalla
        ff=figure('Units','normalized','Name','BOTONES') ;

        hold ;
        kil=0.0;

```

```

ylabel('Roll & Pitch tab'); axis([-20 30 -30 40]);
roll=fill([-15 -15 -3 -3],[-2 2 2 -2],[0.5 .5 0]);
pich=fill([17 17 29 29],[-2 2 2 -2],[.5 .5 0]);
locity=fill([1 1 13 13],[-10 10 10 -10],[0.5 0 0.5]);
% cuad=fill([10 10 16 16],[20 25 25 20],[.5 0 0.5]);
vel=uicontrol('Style','text','Units','normalized',...
    'Position',[0.5 0.65 0.1 0.1],'HorizontalAlignment','center',...
    'BackgroundColor',[1 1
1], 'FontSize',20, 'FontWeight','bold','ForegroundColor',[0 0 0],...
    'String',sprintf('V : %1.2f',0.0));
rolln=uicontrol('Style','text','Units','normalized',...
    'Position',[0.20 0.3 0.05
0.05], 'HorizontalAlignment','center',...
    'BackgroundColor',[1 1
1], 'FontSize',14, 'FontWeight','bold','ForegroundColor',[0 0 0],...
    'String',sprintf('Roll : %1.1f',0.0));
pitchn=uicontrol('Style','text','Units','normalized',...
    'Position',[0.80 0.3 0.05
0.05], 'HorizontalAlignment','center',...
    'BackgroundColor',[1 1
1], 'FontSize',14, 'FontWeight','bold','ForegroundColor',[0 0 0],...
    'String',sprintf('Pitch : %1.1f',0.0));
marca=fill([-15 -15 -5 -5],[10 30 30 10],[1 0 0]);
marca1=fill([-15 -10 -5 -10],[20 30 20 10],[1 0.8 0]);
marca2=fill([-12.5 -10 -7.5 -10],[20 25 20 15],[0 1 0]);

asterix=plot(-10,20,'*')
asterix.Color=[0 0 0];

% R = .6;
% tita = (0:0.01:2.01*pi);
% x = R*cos(tita);
% y = R*sin(tita);
%
% plot(x,y,'-b')
% axis equal
% axis([-1.5 1.5 -1.5 1.5])
% set(gca,'XTick',[-1.5:0.5:1.5])
% set(gca,'XTickLabel',{'-1.5','-1.0','-0.5','0.0','0.5','1.0','1.5'})
% set(gca,'YTick',[-1.5:0.5:1.5])
% set(gca,'YTickLabel',{'-1.5','-1.0','-0.5','0.0','0.5','1.0','1.5'})
% set(gca,'FontSize',10)
%estimada=fill([-1,0,1],[

% Creación de objetos

%obp=fill([1.6 1.6 2 2.4 2.4],[64.404 88.404 95.404 88.404
64.404],[0.5 0.5 0.5]) ; % Creación capsula

% ¿ velocidad=get(v,'Value'),?

bb_str=uicontrol('Style','togglebutton','Units','normalized') ;
set(bb_str,'Position',[0.15 0.1 0.1 0.1],'String','Start') ;
% Esto asigna un manejador a la pulsación del botón. Cuando se
pulsas

```

```

    % hace una llamada con flag=11 (que no lo usa Simulink) para que
    % gestione la pulsación. Tan solo hay que poner un flag diferente
para
    % cada botón
    set(bb_str, 'Callback', 'sfun_botones([], [], [], 11)') ;

    bb_stp=uicontrol('Style', 'pushbutton', 'Units', 'normalized') ;
    set(bb_stp, 'Position', [0.65 0.1 0.1 0.1], 'String', 'Stop') ;
    % El manejador del este botón está en flag=12
    set(bb_stp, 'Callback', 'sfun_botones([], [], [], 12)') ;

%
sld_kg=uicontrol('Parent', ff, 'Style', 'slider', 'Units', 'normalized', ...
    'Position', [0.5 0.5 0.1
0.05], 'Min', 0, 'Max', 180, 'Value', kil, ...
    'Callback', 'ff([], [], [], 'sld_kg')');
%
%
txt_kg=uicontrol('Parent', ff, 'Style', 'text', 'Units', 'normalized', ...
    'Position', [0.5 0.4 .1 0.05], 'HorizontalAlignment', 'center', ...
    'BackgroundColor', [1 1 1], 'ForegroundColor', [0 0 0], ...
    'String', sprintf('KILOS : %1.3f', kil));

    axis off;
    sys=[0 0 0 5 0 0] ; %
Número de entradas : 6
    x0=[];

    case 9,
        close(gcf) ; % Cierra la interfaz

        % AQUI PUEDES LANZAR EL .m QUE TRAZA LAS GRÁFICAS

        sys = [];
        case 11, % Este es el manejador del botón START

            if (get(gco, 'Value')==1)
                set(gco, 'String', 'Pausa') ; % Esto camia el texto del botón
                set_param('Septiembre', 'SimulationCommand', 'Continue') ; %
Esto lanza al Simulink una orden de Continue
            else
                set(gco, 'String', 'Start') ;
                set_param('Septiembre', 'SimulationCommand', 'Pause') ; % Esto
lanza al Simulink una orden de Pause
            end

            case 12, % Este es el manejador del botón STOP
                %kil=get(sld_kg, 'Value');
                %set(txt_kg, 'String', sprintf('Kilos:%1.3f', kil));
                set_param('Septiembte', 'SimulationCommand', 'Stop') ; % Esto lanza
al Simulink una orden de Stop

end

```

sfun_botones.m (SFunction)

```
function [sys,x0,str,ts] = sfun_botones(t,x,u,flag)

global txt

switch flag,

    case 0,

        ff=figure('Units','normalized','Name','BOTONES') ;

        bb_str=uicontrol('Style','togglebutton','Units','normalized') ;
        set(bb_str,'Position',[0.1 0.1 0.3 0.1],'String','Start') ;
        % Esto asigna un manejador a la pulsación del botón. Cuando se
        pulsa
        % hace una llamada con flag=11 (que no lo usa Simulink) para que
        % gestione la pulsación. Tan solo hay que poner un flag diferente
        para
        % cada botón
        set(bb_str,'Callback','sfun_botones([],[],[],11)') ;

        bb_stp=uicontrol('Style','pushbutton','Units','normalized') ;
        set(bb_stp,'Position',[0.6 0.1 0.3 0.1],'String','Stop') ;
        % El manejador del este botón está en flag=12
        set(bb_stp,'Callback','sfun_botones([],[],[],12)') ;

        % Solo pongo esto para que se vea el tiempo corriendo o parado

        txt=uicontrol('Style','text','Units','normalized','BackgroundColor','k
        ',...

        'ForegroundColor','w','HorizontalAlignment','center','FontSize',10,...
        'Position',[0.3 0.5 0.4 0.05],'String',sprintf('t =
        %2.2f',0.0)) ;

        sizes = simsizes;
        sizes.NumContStates = 0;
        sizes.NumDiscStates = 0;
        sizes.NumOutputs = 0;
        sizes.NumInputs = 0;
        sizes.DirFeedthrough = 0;
        sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];
        str = [];
        ts = [0 0];

    case 2,
```

```

    % Esto lanza al Simulink una orden de Pause en el primer paso de
    simulación para que se detenga hasta que se pulse Start
    % Hay que poner el nombre del fichero .slx SIN EXTENSIÓN. En este
    ejemplo el fichero es Botones.slx
    if (t==0) set_param('Septiembre','SimulationCommand','Pause') ;
end ;

set(txt,'String',sprintf('t = %2.2f',t)) ; % Actualiza el texto

sys=[];

case 3,

    sys=[] ;

case 9,
    close(gcf) ; % Cierra la interfaz

    % AQUI PUEDES LANZAR EL .m QUE TRAZA LAS GRÁFICAS
    'Resultados';
    sys = [];

case 11, % Este es el manejador del botón START

    if (get(gcf,'Value')==1)
        set(gcf,'String','Pausa') ; % Esto cambia el texto del botón
        set_param('Septiembre','SimulationCommand','Continue') ; %
Esto lanza al Simulink una orden de Continue
    else
        set(gcf,'String','Start') ;
        set_param('Septiembre','SimulationCommand','Pause') ; % Esto
lanza al Simulink una orden de Pause
    end

case 12, % Este es el manejador del botón STOP

    set_param('Septiembre','SimulationCommand','Stop') ; % Esto lanza
al Simulink una orden de Stop

end

```

sfun_RT.m (SFunction)

```

function [sys,x0,str,ts] = sfun_RT(t,x,u,flag,T)

global t1

switch flag,

    case 0,

        t1=tic ;
        sizes = simsizes;
        sizes.NumContStates = 0;
        sizes.NumDiscStates = 0;

```

```

        sizes.NumOutputs      = 1;
        sizes.NumInputs       = 0;
        sizes.DirFeedthrough = 1;
        sizes.NumSampleTimes = 1;
        sys = simsizes(sizes);
        x0 = [];
        str = [];
        ts = [0 0];

    case 3,
        t2=toc(t1) ;
        if t2>=T
            sys=[1] ;
        else
            sys=[0] ;
            while t2<T
                t2=toc(t1) ;
            end
        end
        t1=tic ;

    case 9,
        sys = [];

end

```

Resultados.m (SFunction)

```

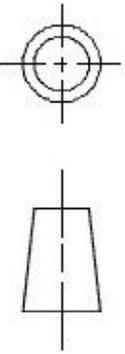
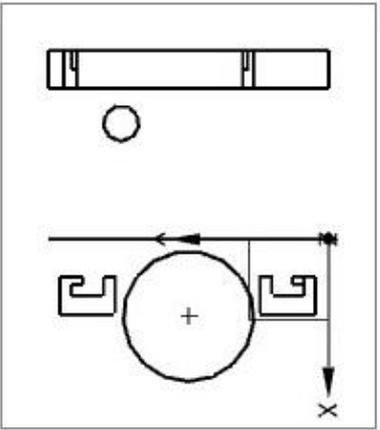
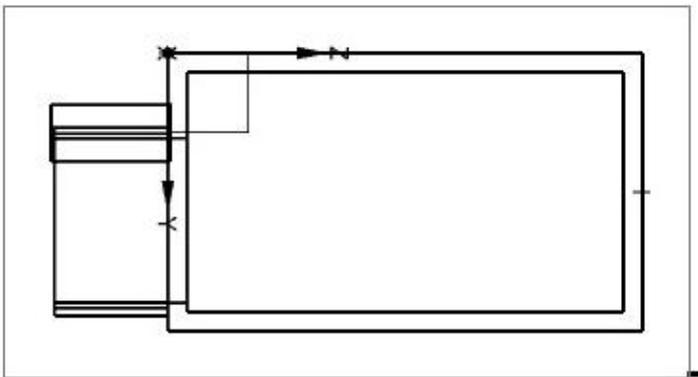
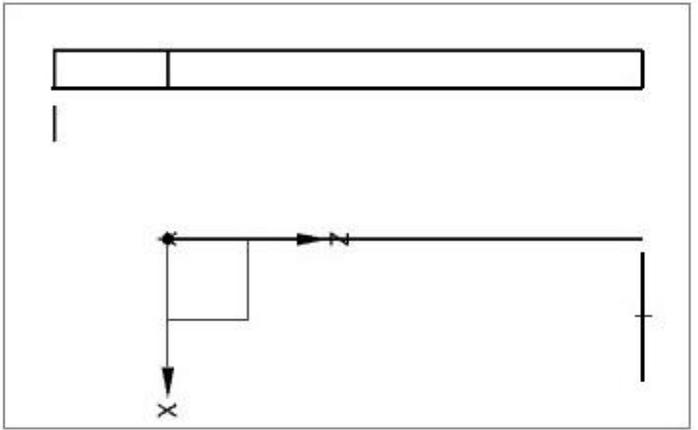
function resultados(peso,velocidad,Roll_Pitch,ddd)
figure (3)
plot(velocidad.time,velocidad.signals(2).values(:,1))
axis([2 ddd.time(end) -0.2 3.5]);
title('Velocidad instantánea')
xlabel('s') % x-axis label
ylabel('m/s') % y-axis label
v_max= max(velocidad.signals(2).values(:,1));
a_max=max(velocidad.signals(1).values(:,1));
F_max=peso*a_max;
pico_potencia_max=peso*v_max;
figure (4);
plot(Roll_Pitch.time,Roll_Pitch.signals(1).values(:,1)/2, 'g')
axis([2 Roll_Pitch.time(end) -30 30]);
title('Ángulo desviación sobre las muñecas')
xlabel('s') % x-axis label
ylabel('°') % y-axis label
figure (5);
plot(Roll_Pitch.time,Roll_Pitch.signals(2).values(:,1), 'g')
axis([2 Roll_Pitch.time(end) -30 30]);
title('Ángulo desviación sobre la horizontal')
xlabel('s') % x-axis label
ylabel('°') % y-axis label
figure (6);
plot(ddd.time,ddd.signals(2).values(:,1))
axis([2 ddd.time(end) 0 500]);
title('Profundidad de repetición')
xlabel('s') % x-axis label

```

```
ylabel('d') % y-axis label
```

```
end
```

Anexo III: Planos



SIEMENS

THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED Pieza superior

DRAWN BY Cristian

CHECKED BY

APPROVED BY

TITLE

SIZE DRG NO.

A4 Sensorpieza1

SHEET REV

A

SCALE 1:2

SHEET 1 OF 1

1

2

3

4

5

A4

ALL DIMENSIONS IN
mm

1

2

3

4

5

6

A

B

C

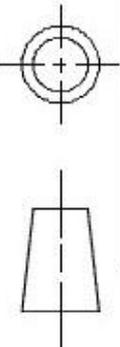
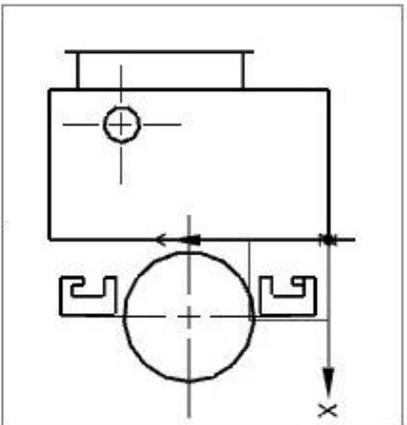
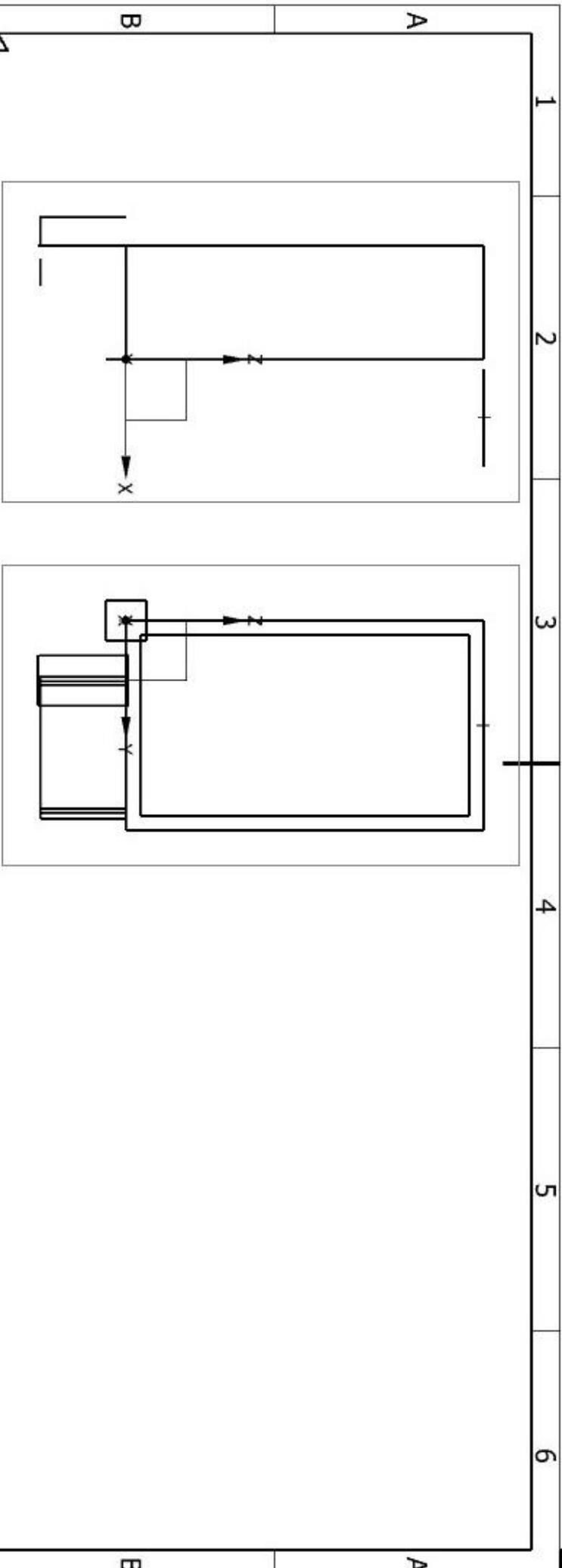
D

A

B

C

D

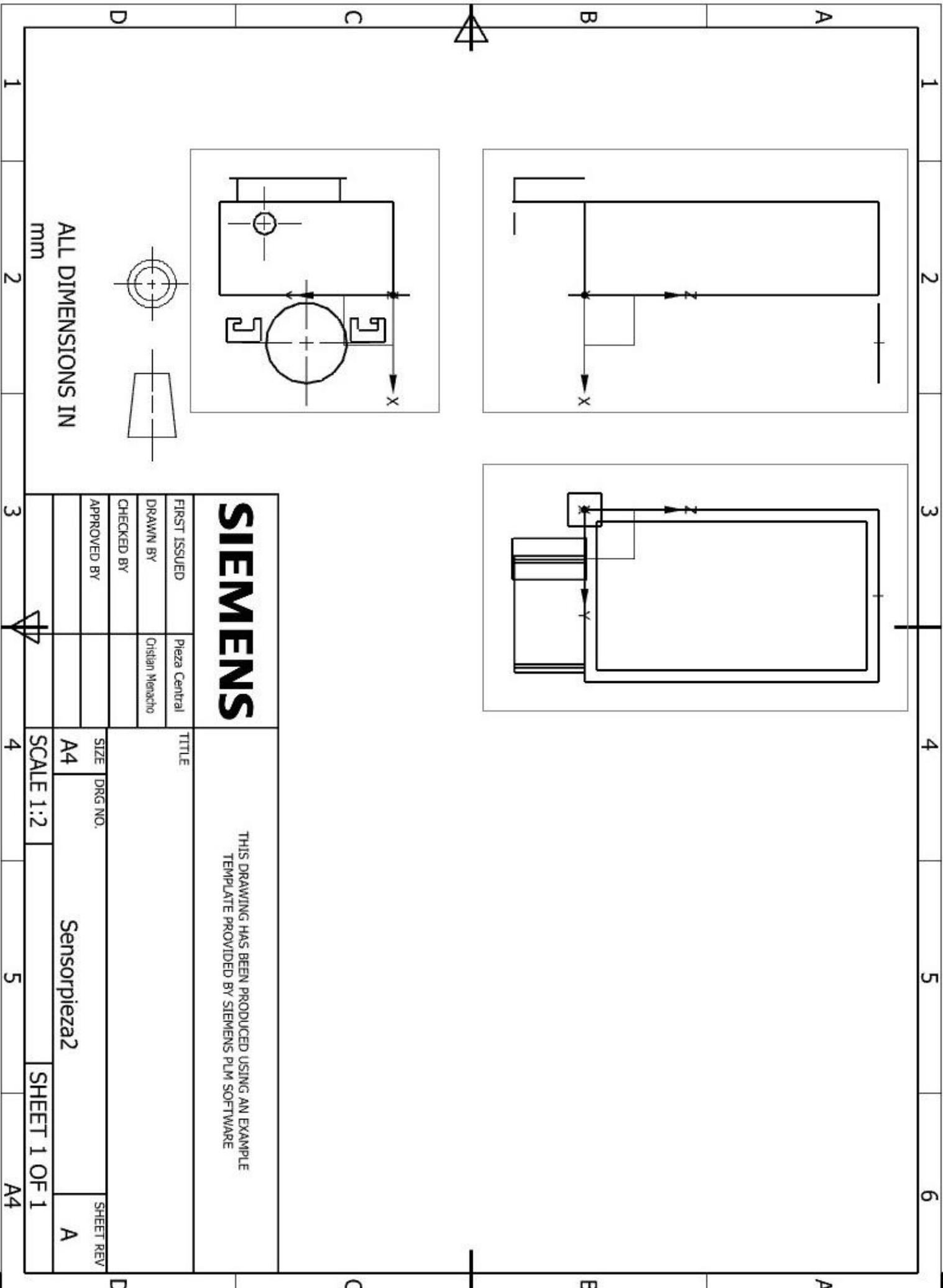


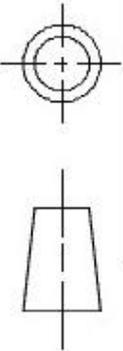
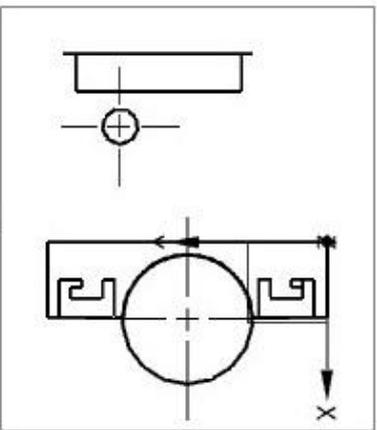
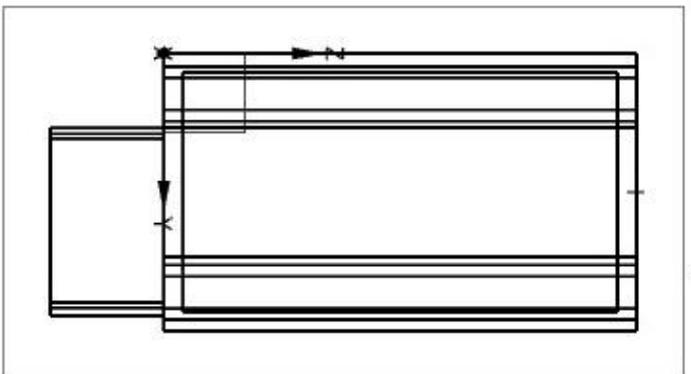
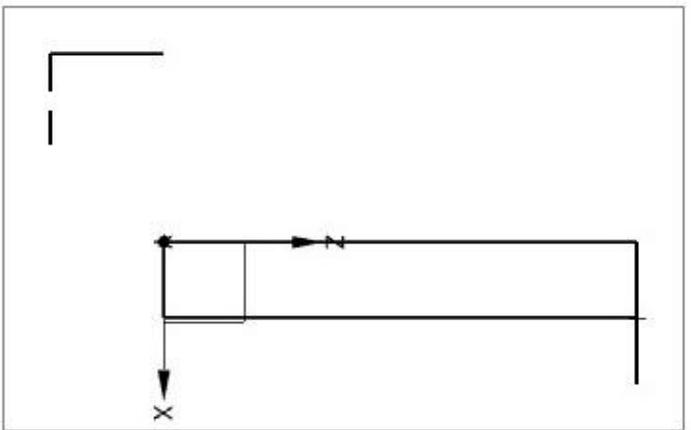
ALL DIMENSIONS IN
mm

SIEMENS

THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED	Pieza Central	TITLE			
DRAWN BY	Cristian Menacho	SIZE	A4	DRG NO.	Sensorpieza2
CHECKED BY		SCALE	1:2	SHEET	1 OF 1
APPROVED BY					
					SHEET REV A





ALL DIMENSIONS IN
mm

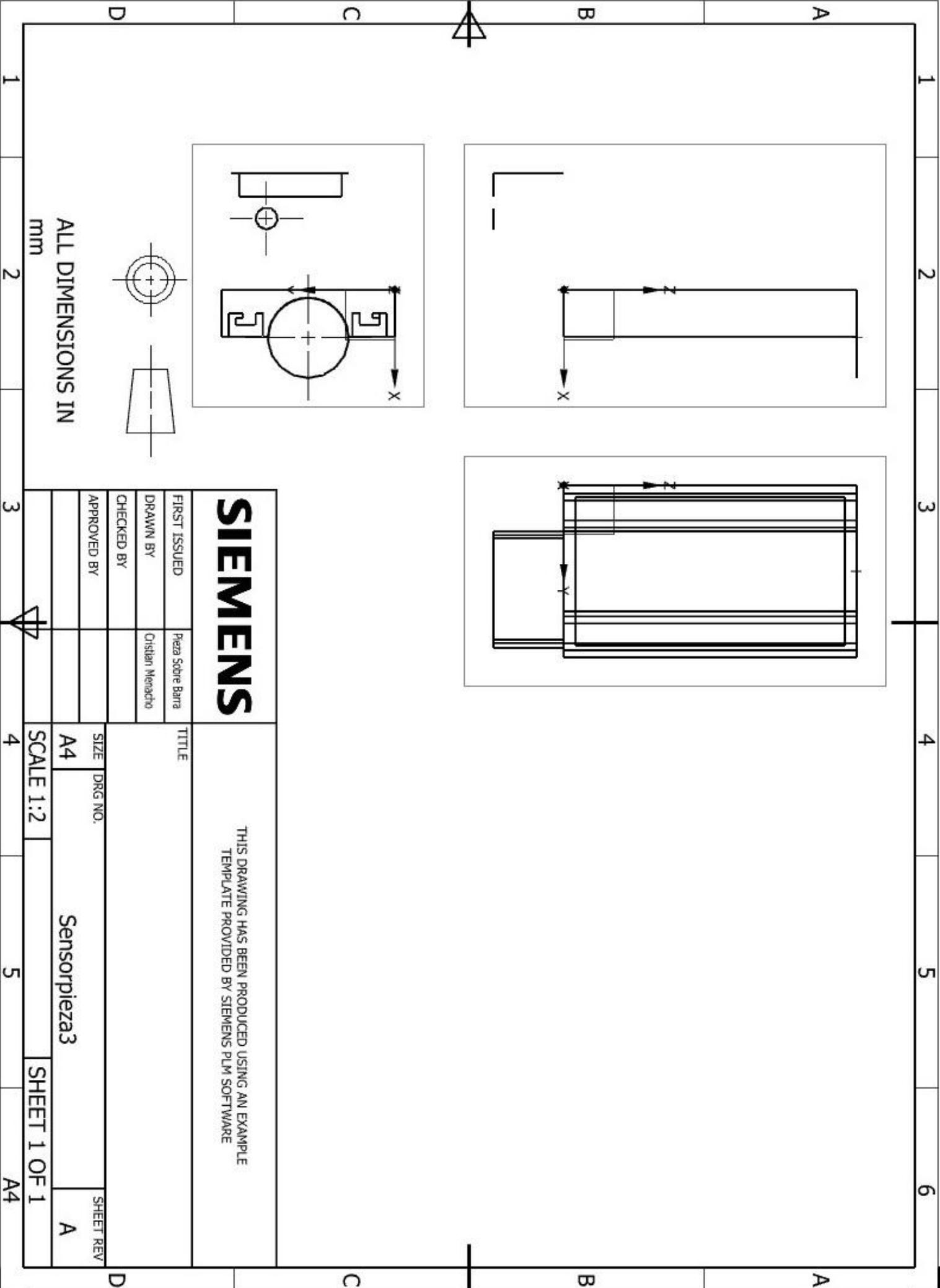
SIEMENS

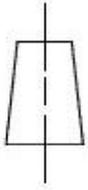
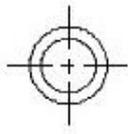
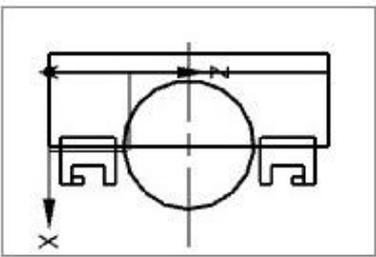
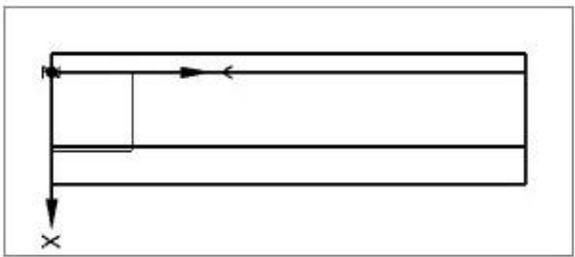
THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED	Pieza Soñre Barra
DRAWN BY	Ostilan Menacho
CHECKED BY	
APPROVED BY	

TITLE		Sensorpieza3		SHEET REV
SIZE	DRG NO.			A
A4				

SCALE 1:2	SHEET 1 OF 1
-----------	--------------





SIEMENS

THIS DRAWING HAS BEEN PRODUCED USING AN EXAMPLE
TEMPLATE PROVIDED BY SIEMENS PLM SOFTWARE

FIRST ISSUED Pieza bajo barra

DRAWN BY Osdira Mendiko Arza

CHECKED BY

APPROVED BY

TITLE

SIZE DRG NO.

A4 Sensorpieza4

SHEET REV

A

SCALE 1:2

SHEET 1 OF 1

ALL DIMENSIONS IN
mm

