

**DISPOSITIVO DIDÁCTICO PARA LABORATORIO DE CINEMÁTICA
CON SISTEMA EMBEBIDO**

**Responsable:
ANDRES FELIPE PALACIOS BAJONERO**

**UNIVERSIDAD PEDAGÓGICA NACIONAL
Facultad de Ciencia y Tecnología
Departamento de Tecnología
Bogotá D.C
2015**

**DISPOSITIVO DIDÁCTICO PARA LABORATORIO DE CINEMÁTICA
CON SISTEMA EMBEBIDO.**

**Responsable:
ANDRES FELIPE PALACIOS BAJONERO**

**Proyecto presentado al Departamento de Ciencia y Tecnología
En la Licenciatura en Electrónica
De la Universidad Pedagógica Nacional
Para la obtención del título de licenciado en electrónica.**

**Director
Diego Fernando Quiroga
Licenciado en electrónica**

**UNIVERSIDAD PEDAGÓGICA NACIONAL
Facultad de Ciencia y Tecnología
Departamento de Tecnología
Bogotá D.C
2015**

Con el fin de enriquecer las prácticas de laboratorio de movimiento parabólico en física cinemática, es necesario poder ajustar las variables iniciales del desplazamiento de un proyectil y lograr recoger los datos pertinentes de su trayectoria. Para cumplir con estas necesidades se diseñó y construyó un dispositivo didáctico para laboratorio de cinemática con un sistema embebido, que facilita el proceso de entender y comprender este tema. El dispositivo cuenta con una estructura adecuada para la experimentación que está formada por un sistema de disparo, que le permite al usuario ajustar la altura, el ángulo y la potencia del disparador, además cuenta con una recámara para observar el desplazamiento del proyectil. El segundo elemento es un software de captura y procesamiento de imagen, que está programado en cross compiler desde Simulink para Raspberry Pi, y cuenta con una interfaz gráfica de usuario (GUI) para la entrega de la información de la experimentación. Este dispositivo constituye un material de apoyo, el cual permite la presentación de la información, la manipulación de las variables, la observación del proceso y el análisis de los resultados. Sin embargo, se presentaron limitaciones a la hora de la experimentación debido a la construcción de la estructura y el empalme de los sistemas con los cuales se trabajó. Aunque, a partir de la experimentación se hallaron otros posibles usos para este proyecto que no se limitan a la física cinemática.

To enrich laboratory practice of kinematic, specifically parabolic motion of projectiles in physics, it is necessary to adjust the variables of the trajectory of a projectile and to get the information of its trajectory. To meet these needs, it was designed and built a didactic laboratory device, which has an embedded system. This didactic device helps the students in the process of learning this topic. The device has a suitable structure for experimentation. It has a trigger system that allows users to adjust the height, the angle and the power of the trigger. This device also has an enclosed space to observe the displacement of the projectile. This device not only has a structure but also a software that captures and processes the image of the projectile triggered. This software is programmed using *cross compiler* from the application *Simulink*. The application is stored in the Raspberry Pi or a little computer and a GUI shows the information that is obtained from the experimentation.

This device is a supporting material that presents the information, allows adjusting the variables as well as the observation process and the analysis of the results. Nevertheless, during the testing process of the structure some limitations were presented. There were also some others problems or limitations with the connection of the systems with which we worked. Although, from experimentation, we found out other uses that could not be related with kinematics.

RESUMEN ANALÍTICO EN EDUCACIÓN - RAE

1. Información General	
Tipo de documento	Proyecto de grado
Acceso al documento	Universidad Pedagógica Nacional. Biblioteca Central
Título del documento	Dispositivo didáctico para laboratorio de cinemática con sistema embebido
Autor(es)	Palacios Bajonero, Andres Felipe
Director	Quiroga , Diego Fernando
Publicación	Bogotá, Universidad Pedagógica Nacional, 2015, 106 p
Unidad Patrocinante	Universidad Pedagógica Nacional
Palabras Claves	DISPOSITIVO, DIDÁCTICO, LABORATORIO, CINEMÁTICA, SISTEMA EMBEBIDO.

2. Descripción
<p>El trabajo tiene como fin principal, diseñar y construir un dispositivo didáctico para laboratorio de cinemática con sistema embebido, que se constituye como una herramienta didáctica de apoyo, para los estudiantes y el docente durante el proceso de enseñanza aprendizaje.</p>

3. Fuentes
<p>Las fuentes utilizadas para el proyecto de grado fueron 19, de las cuales, las más relevantes son las siguientes:</p> <p>[1] C. Platero, “Apuntes de Visión Artificial”. Consultado Octubre de 2015. Disponible en: http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf [2] G. González; C. Eduardo (2008). Los métodos de enseñanza en el proceso de enseñanza aprendizaje de las Ciencias Exactas. V Congreso Internacional Didáctica de las Ciencias, Ciudad de la Habana.</p> <p>[2] J. Jiménez; B. Canacue. (2012). Kit didáctico de laboratorio para la enseñanza de movimiento unidimensional y planos inclinados con fricción. Universidad Pedagógica Nacional, Bogotá, Colombia.</p>

- [3] J.I. Pozo. (1989). Teorías cognitivas del aprendizaje. Morata. España, Madrid.
- [4] J.I. Pozo. (1996). Aprendices y maestros. La nueva cultura del aprendizaje. Alianza. España, Madrid.
- [5] Morales, A. Briceño, E. & Mena, J. (2011). El rol de las gráficas en la construcción de modelos. Pontificia Universidad Católica de Valparaíso, Chile
- [6] R. E. Woods; R. C. González; S. L. Eddins. (2009). Digital image processing using Matlab. Gonzalez, Woods, & Eddins en: F. Giraldo; M. Gonzales; E. Camargo. (2011). Algoritmos de procesamiento de imagen satelitales con transformada Hough. Revista Visión Electrónica.
- [7] Raymond, A. Serway & John, W. Jewett, Jr. (2008). Física para ciencias e ingenierías 7° edición. Cengage Learning Editores, México, D.F
- [8] T. Acharya; A. K. Ray. (2005). Image processing: principles and applications. John Wiley & Sons en: F. Giraldo; M. Gonzales; E. Camargo. (2011). Algoritmos de procesamiento de imagen satelitales con transformada Hough.
- [9] Tipler, M. (2001). Física para la ciencia y la tecnología 5° Edición. Editorial reverté. España, Barcelona.
- [10] H. Youg; G. Freedman. (2009). Física Universitaria 10° Edición. Person Educación, México, D.F

4. Contenidos

El trabajo se divide en cinco capítulos que no incluyen conclusiones, bibliografía y los anexos que contienen procesos de desarrollo del proyecto. A continuación se mencionan los capítulos:

Descripción del problema: En el cual se plantea, delimita y justifica la razón para el desarrollo del presente trabajo, junto a los objetivos principales del mismo.

El marco teórico: En el cual se recopila toda la información que fundamenta el desarrollo del trabajo.

Desarrollo del proyecto: Se dedica totalmente a la descripción sobre el diseño y construcción del dispositivo, donde se enuncian dos etapas, las cuales son: primera etapa, descripción de la estructura y segunda etapa, descripción del software.

Pruebas y resultados: Se describe el desarrollo de las pruebas al dispositivo y los resultados obtenidos de los mismos. **Proyecciones:** Se presenta los alcances obtenidos por el dispositivo y las posibles mejoras y aplicaciones a futuro. **Guía de usuario:** En el cual se presenta una explicación detallada del uso apropiado del dispositivo didáctico para laboratorio de cinemática con sistema embebido.

5. Metodología

Se utilizó una metodología de prototipado para desarrollar y construir la estructura y el software, y una metodología de construcción para desarrollar el manual del funcionamiento del dispositivo didáctico para laboratorio de cinemática con sistema embebido.

6. Conclusiones

1. En primer lugar, se puede concluir que se logró e implementar una estructura que cumple con las condiciones apropiadas para desarrollar la experimentación en física cinemática. A su vez, se logró desarrollar e implementar un software para la adquisición de imagen y reconocimiento del objeto, del cual se calcula la información pertinente para la experimentación y entrega al usuario los datos de tal experimentación en una GUI.
2. Este dispositivo constituye un material de apoyo, el cual facilita la presentación de la información, la manipulación de las variables, la observación del proceso y el análisis de los resultados.

3. Es importante hacer pruebas para calibrar los diferentes sistemas mecánicos que son accionados por motores, dado que si no tienen la suficiente fuerza o tiene más de la requerida para realizar la acción, puede generar un mal funcionamiento de todo el sistema.
4. Al realizar las experimentaciones se evidenció que la realización de prácticas a pequeña escala exigen equipos de mayor precisión, por lo tanto las pruebas realizadas en el proceso de captura y procesamiento de información son poco confiables. Para mejorar las prácticas de laboratorio es pertinente mejorar los equipos de captura de información o hacer la experimentación a mayor escala.
5. Al trabajar con una metodología de prototipado, es necesario tener los elementos para realizar pruebas y cambios en los sistemas que se estén desarrollando, dado que siempre aparecerán inconvenientes a la hora de la construcción o empalme con otros sistemas.
6. A partir de las experimentaciones realizadas con este trabajo se identificaron otros posibles usos de este dispositivo en la práctica como lo son: el movimiento en una dimensión (eje horizontal), choque elástico y caída libre con sucesivos rebotes.

Elaborado por:	Palacios Bajonero Andres Felipe
Revisado por:	Quiroga Diego Fernando

Fecha de elaboración del Resumen:	30	11	2015
--	----	----	------

Contenido

1.	INTRODUCCIÓN	15
2.	DESCRIPCIÓN DEL PROBLEMA	17
2.1	Planteamiento del Problema - Justificación	17
2.2	Objetivos	18
2.1.1	Objetivo General	18
2.1.2	Objetivos Específicos	18
2.3	Antecedentes	19
3.	MARCO TEÓRICO	21
3.1	Cinemática	22
3.1.1	Movimiento en una dimensión	23
3.1.2	Movimiento en dos Dimensiones	29
3.2	VISIÓN ARTIFICIAL	30
3.2.1	Introducción a la Visión Artificial	30
3.2.2	Procesamiento de imagen	31
3.3	COMPONENTES ELECTRÓNICOS	35
3.3.1	ARDUINO	35
3.3.2	RASPBERRY PI B	37
3.3.3	WEBCAM PHILIPS SPC1300NC	38
3.3.4	SERVOMOTOR DONGBU ROBOT HERKULEX DRS-0101	40
4.	DESARROLLO DEL PROYECTO	41
4.1	DESCRIPCIÓN DE LA ESTRUCTURA	42
4.1.1	DISPARADOR	45
4.1.2	AJUSTE DE ÁNGULO	47
4.1.3	AJUSTE DE ALTURA	49
4.1.4	AUTO RECARGA	54

4.1.5	CONTROLES DE ESTRUCTURA	58
4.2	DESCRIPCIÓN DEL SOFTWARE	61
4.2.1	CAPTURA DE IMAGEN	62
4.2.2	PROCESAMIENTO DE DATOS.....	70
5	PRUEBAS Y RESULTADOS	77
6	GUÍA DE USUARIO	88
7	PROYECCIONES.....	90
8	CONCLUSIONES.....	93
9	BIBLIOGRAFÍA.....	94

LISTA DE FIGURAS

Figura 1. Movimiento de una partícula con velocidad constante. (J. Jiménez; B. Canacue. (2012). Pág. 19)

Figura 2. Velocidad instantánea.(J. Jiménez; B. Canacue. (2012). Pág. 20)

Figura 3. Trayectoria parabólica de un proyectil. (Serway- Jewett, 2008. Pág. 77)

Figura 4. Componentes del Arduino Uno

Figura 5. Raspberry Pi modelo B

Figura 6. Webcam Philips SPC1300NC

Figura 7. Servomotor dongbu robot herkulex DRS-0101

Figura 8. Ventada de captura de información.

Figura 9. Foto frontal de la estructura completa de experimentación.

Figura 10. Dirección de movimiento de los sistemas que posee la estructura de

Figura 11. Pieza 1 y pieza 2 del sistema de disparo.

Figura 12. Sistema de disparo

Figura 13. Unión entre la plataforma y el tornillo sin fin.

Figura 14. Parte superior del sistema de ajuste de ángulo.

Figura 15. Parte frontal del sistema de ajuste de ángulo.

Figura 16. Corona plástica y unión

Figura 17. Estructura de coronas para movimiento del motor y el sistema de ajuste de altura

Figura 18. Estructura mecánica para el ajuste de altura de la plataforma de disparo

Figura 19. Unión de las piezas para el tornillo sin fin.

Figura 20. Tornillo sin fin armado con la unión plástica para el ajuste de altura.

Figura 21. Pieza de la unión para el soporte de la plataforma.

Figura 22. Soportes de la plataforma para facilitar el desplazamiento.

Figura 23. Motor con encoder. Referencia 135 485

Figura 24. Unión plástica. Referencia 35 073

Figura 25. Corona dentada plástica. Referencia 35 061

Figura 26. Unión plástica. Referencia 32 064

Figura 27. Unión plástica. Referencia 32 881

Figura 28. Unión plástica con relieve torneado. Referencia 37 926

Figura 29. Unión plástica con relieve torneado y unión para broche. Referencia 37 858

Figura 30. Broche plástico. Referencia 31 915

Figura 31. Broche plástico. Referencia 37 679

Figura 32. Unión plástica con centro torneado. Referencia 37 925

Figura 33. Unión plástica. Referencia 31 426

Figura 34. Unión plástica. Referencia 38 423

Figura 35. Eje superior del sistema de recarga conectado a la cadena plástica y plataforma para el proyectil.

Figura 36. Eje inferior conectado al motor, junto con la cadena plástica y plataforma para el proyectil.

Figura 37. Armado de cadena plástica.

Figura 38. Motor. Referencia 135 485

Figura 39. Unión plástica con corona dentada. Referencia 35 945

Figura 40. Unión plástica. Referencia 35 065

Figura 41. Unión plástica. Referencia 31 983

Figura 42. Unión plástica. Referencia 32 879

Figura 43. Unión plástica para cadena. Referencia 31 762

Figura 44. Unión plástica con inclinación de 7.5° . Referencia 32 071

Figura 45. Pieza de armado de cadena plástica. Referencia 36 248

Figura 46. Controles de los sistemas de la estructura para la experimentación.

Figura 47. Componentes electrónicos para el control de la estructura.

Figura 48. Conexión de bloques de captura de imagen por webcam y salida de imagen por monitor.

Figura 49. Imagen a la salida del bloque de video.

Figura 50. Conexión de bloques de captura de imagen por webcam y salida de imagen por monitor, y bloques de conversión a grises y reconversión a RGB.

Figura 51. Imagen a la salida del bloque de video.

Figura 52. Conexión de bloques de captura de imagen y salida, bloques de conversión y reconversión, y prueba de Matlab Function.

Figura 53. Imagen a la salida del bloque de video.

Figura 54. Proceso de captura y muestra de imágenes con reconocimiento de bordes rectangulares.

Figura 55. Identificación de rectángulos con un libro que posee esta forma.

Figura 56. Proceso de captura y muestra de imágenes con reconocimiento de bordes circulares.

Figura 57. Identificación de círculos utilizando dos datos.

Figura 58. Tratamiento de imagen con el bloque Matlab Function.

Figura 59. Imagen de salida del bloque de video con reconocimiento de color rojo.

Figura 60. Reconocimiento de color rojo sin imagen de fondo.

Figura 61. Captura y tratamiento de imagen con el bloque Matlab Function y envió de datos a Workspace.

Figura 62. Imagen de Workspace con los datos entregados por Simulink.

Figura 63. Sistema de bloques en Simulink para captura y procesamiento de imagen.

Figura 64. Ventana con protocolos utilizados por Matlab y la Raspberry Pi.

Figura 65. Sistema de bloques en Simulink para el procesamiento y entrega de datos.

Figura 66. GUI sin ejecutar datos.

Figura 67. GUI con la tabla de posición del proyectil.

Figura 68. GUI con todos los datos en pantalla.

Figura 69. Reproductor de video de Matlab, para mostrar el desplazamiento del proyectil guardado en una matriz.

Figura 70. Pruebas 1 a la 7, aumento en la velocidad de desplazamiento del proyectil.

Figura 71. Desplazamiento del proyectil respecto al punto de impacto al aumentar la potencia de disparo en las pruebas 36 a la 42.

Figura 72. Desplazamiento con un nivel 3 en el disparador.

Figura 73. Desplazamiento con un nivel 7 en el disparador.

Figura 74. Desplazamiento del proyectil con un nivel 3 en el disparador a 30 cm de la superficie y a 65°.

Figura 75. Disparo a 30°

Figura 76. Disparo a 60°

LISTA DE TABLAS

Tabla 1. Características del Arduino Uno

Tabla 2. Características técnicas de la Raspberry Pi

Tabla 3. Características webcam Philips SPC1300NC

Tabla 4. Características servomotor dongbu robot herkulex DRS-0101

Tabla 5. Pruebas de disparo haciendo variaciones en ángulo del disparador, altura de plataforma de disparo y potencia de disparo.

1. INTRODUCCIÓN

Este documento presenta el diseño de un dispositivo didáctico para laboratorio de cinemática con sistema embebido. Este dispositivo funciona como una herramienta que permite una práctica más enriquecedora para los estudiantes de 9° y 10° grado, a partir de la experimentación con un cuerpo en movimiento.

Para el desarrollo de este trabajo se tuvieron en cuenta tres antecedentes teóricos, a partir de los cuales, se determina la importancia de controlar los factores que pueden alterar la toma de datos y determinar la importancia en la clasificación de los mismos. Para la comprensión y análisis de los datos es importante conocer los conceptos de cinemática, visión artificial y conocer las características de los componentes electrónicos a usar. Dentro de la cinemática se especifica los conceptos de movimiento unidimensional y bidimensional, ya que los datos obtenidos se expresan según estos referentes. Con respecto a la visión artificial se explica en qué consiste y como se procesan las imágenes. En los referentes teóricos, finalmente se encuentran la descripción de los componentes electrónicos que permiten este trabajo, como lo son: el arduino, la Raspberry Pi, la webcam y el servomotor.

En el cuerpo del proyecto se describe la estructura, la cual tiene un sistema de disparo que controla las condiciones iniciales para un movimiento parabólico. Dentro de esta estructura se encuentran componentes que impulsan y miden el ángulo de disparo, el ajuste de altura y permiten la auto recarga de proyectiles. Con respecto al diseño del software, se

ha realizado la captura de imágenes, el reconocimiento de color y coordenadas, así como el control de variables.

En cuanto a las pruebas y los resultados del proyecto, en el punto cinco, se muestra cómo funciona cada una de las partes de la estructura a partir de los resultados de 68 pruebas de disparo, en las que además se describe el tipo de movimiento obtenido según las variaciones de altura, ángulo y potencia de disparo. Esta etapa del proyecto termina con la prueba de la Interfaz Gráfica de Usuario (GUI) desde donde se manipulan los diferentes sistemas de procesamiento.

Este proyecto incluye una guía de usuario la cual permite dar conocer los componentes de la estructura y su funcionamiento de manera tal que sea fácil para el usuario poder manejar el sistema. Posteriormente, se evidencian los avances y limitaciones del proyecto, así como una posible solución a los problemas presentados.

Finalmente, se concluye que este dispositivo constituye un material de apoyo, el cual facilita la presentación de la información, la manipulación de las variables, la observación del proceso y el análisis de los resultados. Sin embargo, se presentaron limitaciones a la hora de la experimentación debido a la construcción de la estructura y el empalme de los sistemas con los cuales se trabajó. Aunque, a partir de la experimentación se hallaron otros posibles usos para este proyecto.

2. DESCRIPCIÓN DEL PROBLEMA

2.1 Planteamiento del Problema - Justificación

Con el fin de lograr una práctica más enriquecedora de laboratorio de cinemática para los estudiantes de colegio de 9° y 10° grado, y estudiantes universitarios, se ve la necesidad de construir una herramienta didáctica con sistema embebido, la cual se ubique dentro de un espacio que permita la captura de datos y el procesamiento de los mismos, a través de un ordenador de placa reducida (single board).

Con el objetivo de que el estudiante pueda experimentar el movimiento de un cuerpo, dentro de la propuesta se incluye la adaptación de un espacio para el registro de datos que permita una mayor precisión al disminuir factores ambientales como la luz y el aire que puedan alterar el movimiento del objeto y por ende los datos en la observación. Una posible alternativa para el registro de datos es la visión artificial, ésta facilita la captura de más datos en menor tiempo para aumentar la precisión en la trayectoria del objeto en movimiento.

Esta propuesta permite que el estudiante no sólo observe, sino que pueda modificar algunas variables para realizar diferentes ejercicios. Estas variables son el ángulo y la fuerza con la que se realiza el disparo. Esto se efectuará de forma mecánica utilizando un sistema embebido que además tiene la ventaja de ocupar un menor espacio y estará programado únicamente para la toma de datos de física cinemática. Una posible alternativa de solución al problema del disparador, será utilizado un servomotor que generará el

desplazamiento en grados para ajustar la altura de disparo y un sistema de aire a presión ajustable, para aplicarle fuerza al proyectil.

Es pertinente la realización de este trabajo dado que tiene dos componentes, uno técnico, al diseñar y construir un software y dispositivo que cumpla una finalidad específica, y un componente pedagógico al generar una herramienta que ayude tanto al docente como al estudiante a construir y profundizar en un campo del saber cómo es la cinemática. De tal manera que este dispositivo se convierta en una herramienta didáctica para la experimentación, como recurso didáctico en la enseñanza y el aprendizaje de nuevos conceptos y la puesta en práctica de los mismos. Con el fin de que se logre evidenciar los conocimientos adquiridos en la licenciatura en electrónica de la Universidad Pedagógica Nacional.

2.2 Objetivos

2.1.1 Objetivo General

Diseñar y construir un dispositivo de experimentación para laboratorio de cinemática el cual se apoye en un sistema embebido con sistema operativo.

2.1.2 Objetivos Específicos

Elaborar el control de un disparador que maneje posición, fuerza de disparo y recarga automática del proyectil, dentro de un espacio con las condiciones apropiadas para la experimentación.

Diseñar un software que permita la adquisición de datos por medio de sensores, el procesamiento de los mismos y su visualización en una pantalla.

2.3 Antecedentes

A continuación se presentan tres documentos, entre los cuales hay trabajos de grado de universidades de Chile, México y España, donde se han propuesto herramientas didácticas en el área de física. El primero es un sistema de seguimiento visual de objetos, el segundo desarrolla guías de trabajo en laboratorio y por último, se encuentra una monografía que explica el beneficio de las gráficas en el proceso de modelación.

El primer documento que se reseña es el de Sánchez C. Ramos, el cual está encaminado al seguimiento visual de objetos utilizando técnicas de predicción. Una de las técnicas de la predicción es la estereo visión, a través de la cual se detectan objetos y se predice la posición futura utilizando modelos matemáticos. Como resultado se obtuvo que al manejar diferentes cámaras se podía tener dificultad al tomar los datos debido a iluminación, movimiento de las cámaras por el viento, fricción del aire con el objeto o sincronización de la cámara con el ordenador. (Sánchez. 2010.)

El segundo trabajo que se presenta es el de Martin D. Castillo Sánchez, el cual está enfocado en prácticas de laboratorio en secundaria de las asignaturas de física I y física II. A partir de diferentes materiales didácticos el estudiante fortalece de manera experimental los conceptos trabajados en el área de física. Para llevar a cabo este proyecto se utilizó la teoría de aprendizaje significativo con el fin de generar motivación, innovación y

autonomía. El autor concluye que los alumnos mostraban un mayor interés al utilizar el material didáctico en los laboratorios y se evidenciaba en el aumento de la participación e inclusión de otros materiales a las prácticas. (Pons. 2010)

El último texto es el de Astrid Morales Soto, el cual está orientado al rol de las gráficas en la construcción de modelos. A través de imágenes de graficas de todas las soluciones posibles a un problema, el estudiante indaga cuál puede ser la respuesta más acertada, valiéndose de todas las ecuaciones posibles en ese tema, para luego socializar y cuestionarse acerca de cuál era la correcta o la que se acercaba más a la solución. La metodología empleada en este trabajo es cualitativa, basada en entrevistas en torno a un diseño de situación planteado. Se logró evidenciar dos cosas importantes: la necesidad de las gráficas como argumento para explicar un fenómeno, y el permitir la configuración del tipo de relación entre variables. (Morales 2011)

A partir de la lectura de los anteriores textos se pueden tomar algunas consideraciones que permitirán la mejor realización de este trabajo. En primera medida, solventan vacíos en la adquisición de datos por medio de visión artificial, teniendo en cuenta los diferentes factores que pueden alterar la toma de datos; también encontramos una ayuda para formular guías de trabajo de laboratorio alrededor de un tema en específico, como lo es la física cinemática; y por último tenemos un texto sobre la importancia de las gráficas en la construcción de modelos, el cual ayudará en este trabajo a determinar qué datos son de importancia a la hora de mostrar información en este tipo de presentación.

3. MARCO TEÓRICO

El aumento en el desarrollo científico y su importancia a nivel social, está generando que la escuela logre una dinámica a fin a la evolución que ocurre en la sociedad, y que sea vital que se involucre la ciencia y la técnica en la enseñanza, como parte de los procesos educativos. (G. González; C. Eduardo, 2008) En los últimos años, se muestra una constante transformación en los planes y programas de estudio de las disciplinas que se encargan de la enseñanza de las ciencias. Estos planes se enfocan en la experimentación, la cual se define como aquella experiencia científica en la cual se provoca deliberadamente algún cambio o se manipula intencionalmente una acción, para observar, interpretar y analizar sus resultados o los posibles efectos, en otras palabras, es un estudio de investigación en donde se manipula una o más variables independientes, para analizar las consecuencias de esa manipulación sobre una o más variables dependientes, dentro de una situación de control para el investigador. (J.I. Pozo, 1989) (J.I. Pozo, 1996)

Conociendo que es la experimentación, es común afirmar que la educación científica debe estar dirigida a fomentar la construcción de conocimientos por parte de los alumnos, en lugar de meramente repetir o reproducir sistemas de conocimiento ya elaborados. Por tal razón se puede decir que se construyen nuevos conocimientos en los alumnos de una forma significativa, al hacer que interactúen con el medio y traten de comprenderlo, esto genera una asimilación de conceptos más fuerte y más duradera. Con este fin, el presente proyecto consiste en brindar a los estudiantes un material didáctico que facilite el proceso de enseñanza-aprendizaje. Este material permite interpretar la realidad e ilustrar de manera

gráfica aquello que se desea comprender o comunicar (Islas, 2009. Pág. 21), por ejemplo, a partir del dispositivo para laboratorio de cinemática los estudiantes pueden estimular su interés y la comprensión por los temas de física relacionados con la cinemática, además de apropiarse de conceptos propios del área como los que se muestran a continuación.

Para desarrollar el presente trabajo es necesario tener conocimientos previos de algunos temas relacionados con la cinemática, el control de posición, la captura de datos por medio de sensores y sistemas embebidos en general. Para lo que se tomará información de algunos textos relacionados con estos temas.

3.1 Cinemática

En primera instancia debemos hablar de la cinemática, que se define como una parte de la mecánica, que tiene por finalidad describir matemáticamente todos los tipos posibles de movimiento mecánico sin relacionarlo con las causas que determinan cada tipo concreto de movimiento. La cinemática estudia las propiedades geométricas del movimiento, independientemente de las fuerzas aplicadas y de la masa de la partícula (Young y Freedman, 2009). Este es el principal concepto a utilizar dado que el dispositivo propuesto en este proyecto va a estar en función de registrar y analizar los fenómenos físicos cinemáticos presentes en un objeto.

3.1.1 Movimiento en una dimensión

Se considera al movimiento de un objeto a lo largo de una línea recta, movimiento en una dimensión. A partir de las experiencias cotidianas, es claro que el movimiento de un objeto representa un cambio continuo en la posición de sí mismo. El movimiento de un objeto es clasificado en la física en tres tipos: traslacional, rotacional y vibratorio. Un automóvil que viaja en una autopista es un ejemplo de movimiento traslacional, el giro de la Tierra sobre su eje es un ejemplo de movimiento rotacional, y el movimiento de ida y vuelta de un péndulo es un ejemplo de movimiento vibratorio. Para este trabajo, sólo se tendrá en cuenta el movimiento traslacional. En el estudio del movimiento traslacional se usa el modelo de partícula y el objeto en movimiento se describe como una partícula sin importar su tamaño. En general, una partícula es un objeto parecido a un punto, es decir, un objeto que tiene masa pero es de tamaño infinitesimal. Por ejemplo, si quiere describir el movimiento de la Tierra alrededor del Sol, se puede considerar a la Tierra como partícula y obtener datos razonablemente precisos acerca de su órbita (Serway- Jewett, 2008. Pág. 19). Esta aproximación se justifica porque el radio de la órbita de la Tierra es grande en comparación con las dimensiones de la Tierra y del Sol. Como ejemplo en una escala mucho más pequeña, es posible explicar la presión que ejerce un gas sobre las paredes de un contenedor al tratar las moléculas de gas como partículas, sin importar su estructura interna.

3.1.1.1 Velocidad Media

SI se observa el desplazamiento de un objeto a lo largo del eje horizontal, es de nota que hay una diferencia de posición con respecto al tiempo, estas variables conllevan al análisis de la velocidad promedio, o velocidad media, donde x es el desplazamiento, Δx (delta de x), dividida entre el intervalo de tiempo Δt (delta de t) en el que ocurre el desplazamiento (Sears- Zemansky, 2009. Pág. 37). Usamos la variable V_m para representar velocidad media (el subíndice “m” indica que se trata de un valor promedio:

$$V_m = \frac{\Delta x}{\Delta t} = \frac{x_f - x_i}{t_f - t_i} \quad (1)$$

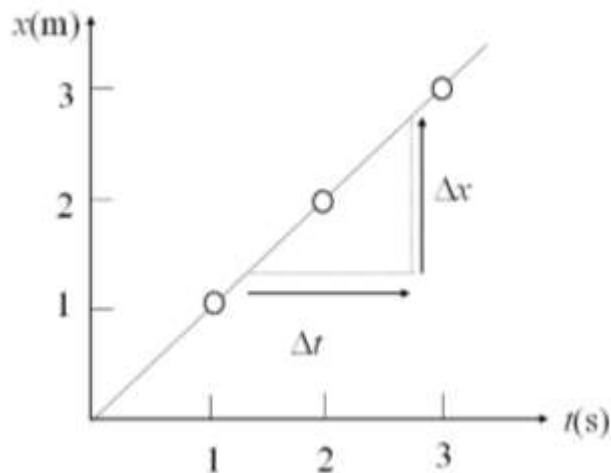


Figura 1. Movimiento de una partícula con velocidad constante. (J. Jiménez; B. Canacue. (2012). Pág. 19)

3.1.1.2 Velocidad Instantánea

La velocidad en un instante se puede definir mediante un proceso de paso de límite. Con la siguiente gráfica (figura 2) se puede entender mejor este concepto. Cuando consideramos sucesivamente intervalos de tiempo más cortos a partir de t_1 o t_i , la velocidad media para cada intervalo se aproxima más a la pendiente de la tangente en t_1 . La pendiente de esta tangente se define como la velocidad instantánea en t_1 . Esta tangente es el límite de la relación $\Delta x/\Delta t$ cuando Δt se aproxima a cero (Tipler, 2001. Pág. 22-23).

$$V_{inst} = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} = \frac{dx}{dt} \quad (2)$$

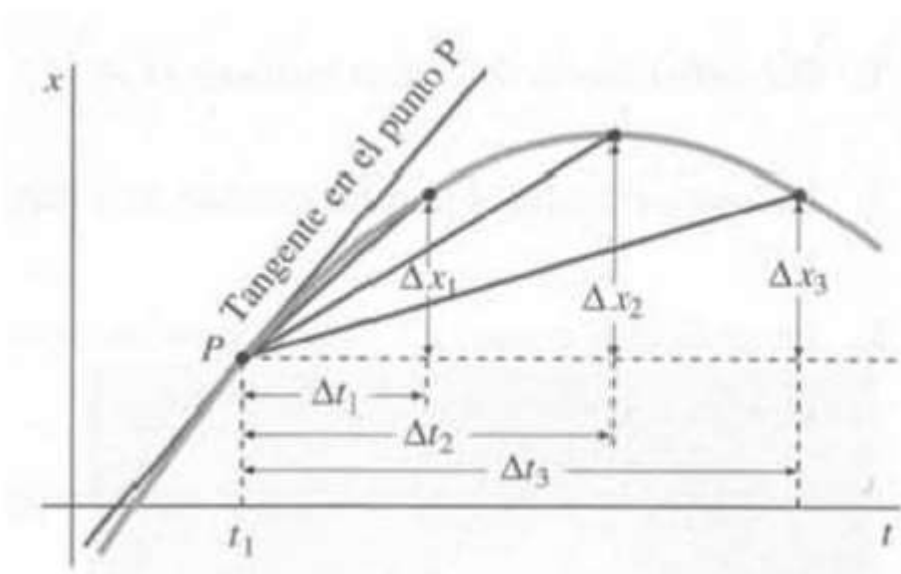


Figura 2. Velocidad instantánea.(J. Jiménez; B. Canacue. (2012). Pág. 20)

3.1.1.3 Aceleración Media

La aceleración es la tasa de cambio de la velocidad instantánea. Cuando, por ejemplo, un conductor aprieta el pedal del acelerador en su coche, espera cambiar su velocidad. La aceleración es un intervalo particular de tiempo (Tipler, 2001. Pág. 25), $\Delta t = t_f - t_i$ donde $t_2 = t_f$ y $t_1 = t_i$, se define como el cociente $\Delta v / \Delta t$ en donde $\Delta v = V_f - V_i$.

$$a_m = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1} \quad (3)$$

3.1.1.4 Aceleración Instantánea

Se puede definir la aceleración instantánea con el mismo procedimiento que se aplicó para la velocidad instantánea. La aceleración instantánea es el límite de la aceleración media conforme el intervalo de tiempo se acerca a cero (Sears- Zemansky, 2009. Pág. 44). En el lenguaje del cálculo, la aceleración es la tasa instantánea de cambio de la velocidad con el tiempo. Así:

$$a_x = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t} = \frac{dv}{dt} \quad (4)$$

3.1.1.5 Aceleración Constante

El movimiento acelerado más sencillo es el rectilíneo con aceleración constante. En este caso la velocidad cambia al mismo ritmo todo el tiempo. Se trata de una situación muy especial, un cuerpo que cae tiene aceleración constante si se desprecia los efectos del aire, lo mismo ocurre con una partícula que se desliza por una pendiente o sobre una superficie horizontal áspera (Tipler, 2001. Pág. 23).

Cuando la aceleración instantánea (a_x) es constante, la aceleración media (a_m) para cualquier intervalo es a . Esto vuelve sencillo derivar las ecuaciones para la posición x , con la velocidad constante como función del tiempo. Para encontrar una expresión, primero sustituimos Δ por la variación para quedar como:

$$a_x = \frac{v_2 - v_1}{t_2 - t_1} \quad (5)$$

Sean ahora $t_1 = 0$ y v cualquier instante posterior t . denotamos con v_0 la componente x de la velocidad en el instante inicial $t = 0$; la componente x de la velocidad en el instante posterior t es V (Sears- Zemansky, 2009. Pág. 48). Entonces, la expresión sería igual a:

$$a = \frac{v - v_0}{t - 0} \quad (6)$$

$V = V_0 + a*t$ solo con aceleración constante en función del tiempo.

Para una aceleración constante, la velocidad varía linealmente con el tiempo y la velocidad media es el valor medio de las velocidades inicial y final, (esta relación es válida solo si la aceleración es constante). Si v_0 es la velocidad inicial y v la velocidad final, la velocidad media es:

$$v_m = \frac{v_0 + v}{2} \quad (7)$$

El desplazamiento es, por tanto,

$$\Delta x = x - x_0 = v_m * t = (v_0 + v) * \frac{t}{2} \quad (8)$$

Podemos eliminar V sustituyendo $V = V_0 + a*t$ de esta manera la expresión sería:

$$\Delta x = (v_0 + v) * \frac{t}{2} = (v_0 + v_0 + a * t) * \frac{t}{2} = v_0 * t + a * \frac{t^2}{2} \quad (9)$$

Con aceleración constante.

El desplazamiento es, así:

$$\Delta x = x - x_0 = v_0 * t + a * \frac{t^2}{2} \quad (10)$$

El término $V_0 * t$ representa el desplazamiento que tendría lugar si fuera cero, y el término $a * \frac{t^2}{2}$ es el desplazamiento adicional debido a la aceleración constante. Eliminado t entre las ecuaciones (6) y (7) se obtiene una expresión entre Δx , a , v y v_0 . De acuerdo a la ecuación (6) $t = (v - v_0)/a$ y sustituyendo en la ecuación (7) se obtiene

$$\Delta x = V_m * t = (V_0 + V) * \frac{t}{2} = \frac{(V_0 + V)(V - V_0)}{2a} = \frac{V^2 - V_0^2}{2a}$$

Es decir:

$$V^2 = V_0^2 + 2a\Delta x \quad (11)$$

3.1.1.6 Caída Libre

El caso más importante del movimiento uniformemente acelerado es el de caída libre bajo la acción de la gravedad. En este caso tomamos la dirección vertical hacia arriba como positiva, definimos $a = -g$, tomando el signo menos debido al hecho de que la aceleración de la gravedad es hacia abajo. El valor de g varía de un lugar a otro de la superficie terrestre, pero es siempre muy cercano a $g = 9.8 \text{ m/s}^2 = 32.2 \text{ ft/s}^2$. Este valor es el mismo para todos los cuerpos, y puede considerarse independiente de la altura, mientras no nos alejemos de la superficie terrestre, ya que la aceleración de la gravedad disminuye a medida que la distancia sobre la superficie terrestre o bajo ella aumenta.

$$V = V_0 + a * t$$

$$\Delta x = x - x_0 = V_m * t = (V_0 + V) * \frac{t}{2}$$

$$\Delta x = x - x_0 = V_0 * t + \frac{at^2}{2}$$

$$V^2 = V_0^2 + 2a\Delta x$$

3.1.2 Movimiento en dos Dimensiones

Conocer el movimiento bidimensional permitirá examinar una diversidad de movimientos, que van desde el movimiento de satélites en órbita al movimiento de electrones en un campo eléctrico uniforme. Para este documento se considera el movimiento de proyectiles como casos especiales de movimiento en dos dimensiones (Serway- Jewett, 2008. Pág. 71).

3.1.2.1 Movimiento de Proyectil

Quien haya observado una pelota de beisbol en movimiento puede decir que observó movimiento de proyectil. La bola se mueve en una trayectoria curva y regresa al suelo. El movimiento de proyectil de un objeto es simple de analizar a partir de dos suposiciones: 1) la aceleración de caída libre es constante en el intervalo de movimiento y se dirige hacia abajo. 2) El efecto de la resistencia del aire es despreciable. Con estas suposiciones, se encuentra que la trayectoria de un proyectil siempre es una parábola, como se muestra en la figura 3 (Serway- Jewett, 2008. Pág. 77). La expresión para el vector de posición del proyectil como función del tiempo se sigue directamente de la ecuación (11):

$$\Delta x = x - x_0 = V_0 * t + \frac{gt^2}{2}$$

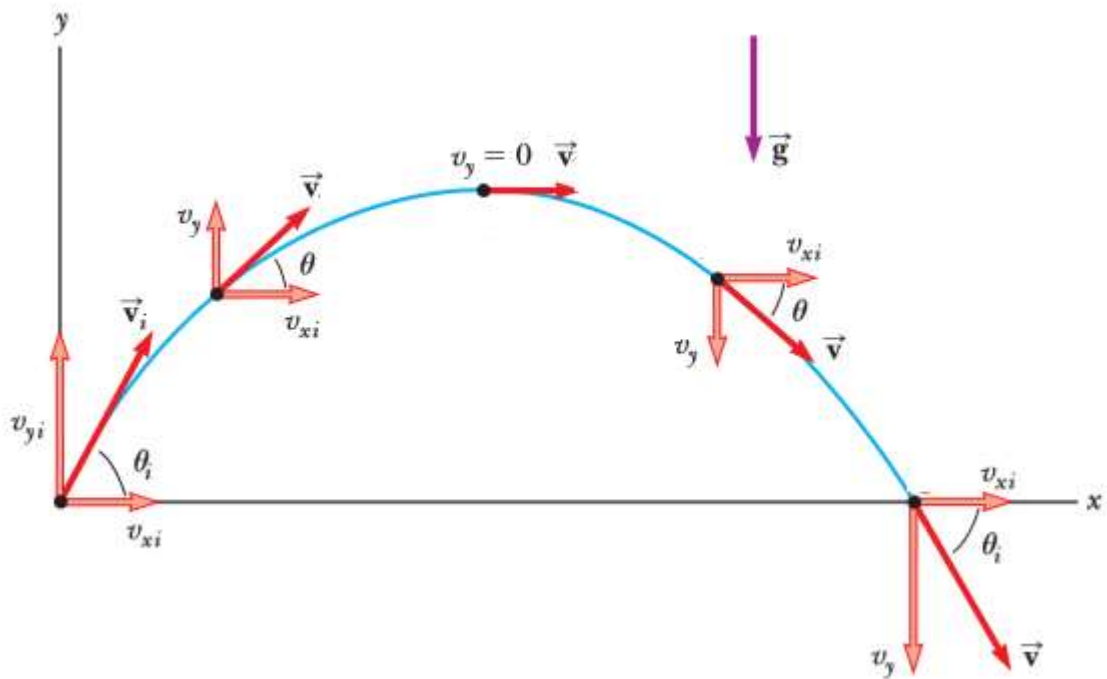


Figura 3. Trayectoria parabólica de un proyectil. (Serway- Jewett, 2008. Pág. 77)

El vector velocidad cambia con el tiempo tanto en magnitud como en dirección. Este cambio es el resultado de la aceleración en la dirección “y” negativa. La componente x de velocidad permanece constante en el tiempo porque no hay aceleración a lo largo de la dirección horizontal. La componente “y” de velocidad es cero en el pico de la trayectoria.

Donde las componentes “x” y “y” de la velocidad inicial del proyectil son:

$$V_{xi} = V_i \cos \Theta_i$$

$$V_{yi} = V_i \sin \Theta_i \quad (12)$$

3.2 Visión Artificial

3.2.1 Introducción a la Visión Artificial

Podríamos decir que la Visión Artificial (VA) describe la deducción automática de la estructura y propiedades de un mundo tridimensional posiblemente dinámico, a partir de

una o varias imágenes bidimensionales del mundo. Las imágenes pueden ser monocromáticas (escalas de gris) o colores, pueden provenir de una o varias cámaras e incluso cada cámara puede ser estacionaria o móvil. Las estructuras y propiedades del mundo tridimensional que queremos deducir en visión artificial incluyen no sólo sus propiedades geométricas, sino también sus propiedades materiales. Ejemplos de propiedades geométricas son: la forma, el tamaño y la localización de los objetos. Ejemplos de propiedades de los materiales son: el color, la iluminación, la textura y la composición. Si el mundo se modifica en el proceso de formación de la imagen, necesitaremos inferir también la naturaleza del cambio, e incluso predecir el futuro. La entrada a un sistema de VA es una imagen obtenida por un elemento de adquisición, mientras que su salida es una descripción de la escena, la cual ha sido obtenida a partir de la imagen. Por un lado, esta descripción debe estar relacionada de algún modo con aquella realidad que produce la imagen y, por el otro, debe contener toda la información requerida para la tarea de interacción con el medio ambiente que se desea llevar a cabo, por ejemplo, mediante un robot.

3.2.2 Procesamiento de Imagen

El procesamiento de imagen o procesamiento digital, se considera el proceso mediante el cual se agrupan varios métodos que partiendo de unas bases matemáticas, plantean la consecución y organización de información respecto a una imagen.

Para desarrollar el procesamiento de las imágenes, es necesario utilizar una serie de técnicas de segmentación, las cuales permiten dividir una imagen, para aislar segmentos, regiones u objetos individuales, que son homogéneos con respecto a algún rasgo, de tal forma se puede extraer características o identificar y clasificar los objetos presentes en la imagen (C. Platero, 2008). Estos datos pueden ser útiles para cuantificar regiones, tomar medidas sobre objetos, discriminar automáticamente los elementos presentes, definir bordes y formas, diferentes tonalidades, etc.

Los criterios generales en los cuales se basan la mayoría de técnicas de para la segmentación de imágenes son:

Similitud: cada uno de los pixeles de un elemento tiene colores parecidos con respecto a alguna propiedad determinada.

Discontinuidad: Los objetos y más concretamente las regiones que los diferencian destacan del entorno y tienen por tanto bordes.

Conectividad: Los pixeles pertenecientes al mismo objeto o región, tienen que ser contiguos, es decir, tienen a agruparse constituyendo regiones homogéneas.

En la segmentación de imágenes en escala de grises, el criterio de discriminación es la intensidad o el tono de gris para buscar agrupar los pixeles de similar comportamiento. En algunas aplicaciones las imágenes pueden binarizarse (umbral dejando dos tonos: blanco y negro) para hacer más sencillo y veloz el procesamiento computacional (Universidad de Jaén, 2008).

En la segmentación de imágenes a color, y si para la aplicación no es relevante la información de color que contiene, se puede simplificar el procesamiento haciendo una conversión a escala de grises para así segmentarla. También se pueden emplear la mayoría de métodos que se han desarrollado para el caso de imágenes en escala de grises, aplicándolos de forma independiente a cada componente de color y procesándolos por separado como si fueran monocromáticos, solo teniendo en cuenta el espacio de color y las cualidades propias del color.

Un color usualmente se puede especificar por la teoría tri-cromática usando tres coordenadas o parámetros, es decir formado por tres luces por separado, roja, verde y azul a partir de las cuales se puede visualizar cualquier color de la escala cromática. Esos tres parámetros definen el color en el denominado espacio de color, que es un modelo que permite especificar, crear y visualizar un color. De acuerdo a la aplicación será más conveniente usar uno u otro espacio de color.

Los espacios de color se dividen en (R.E. Woods; R.C. Gonzáles, 2009):

RGB (Red Green Blue), sistema de color basado en la teoría tri-cromática, es un método que define la cantidad de luz que debe ser emitida para producir un color, cuya fusión recompone la luz blanca. Pero este espacio no es uniforme, en el sentido de que las diferencias de color percibidas como iguales por el ojo humano no corresponden a iguales distancias euclídeas. Este sistema es utilizado por los sensores de las cámaras digitales y tubos de rayos catódicos.

HSL, HSB o HSV, HSI (Hue saturation and Lightness/Brightness/ Intensity) es un sistema que representa intuitivamente el color basado en su percepción, es decir la forma como lo percibe el ser humano. Puede entenderse como una transformación del espacio RGB al espacio perceptual. Posee tres variantes en cuanto a si el color se presenta por luminiscencia, brillo o intensidad.

CIE (CIELuv, CIEUuw and CIELab) es el sistema de color más cercano a la percepción del hombre, se basa en el sistema de medición CIE el cual está relacionado directamente con la visión humana.

CMY (K) (Cyan Magenta Yellow (Black)) es un sistema de color que describe la cantidad de la luz reflejada o sustraída desde la superficie del pigmento, la mezcla de los tres colores produce el negro. Este sistema es utilizado en impresoras y copiadoras.

YIQ, YUV, YCbCr, YCC, YPbPr (Luminance - Chrominance) Estos espacios de color son usados por las trasmisiones de televisión y compresión de imágenes y videos.

Las técnicas clásicas para la segmentación de imágenes se pueden dividir a grandes rasgos en cinco métodos:

- Segmentación basada en el valor del pixel: la forma más simple de segmentar una imagen es en el caso que los pixeles o regiones de una imagen se pueden

diferencias y caracterizar de acuerdo a su escala de gris. El objetivo es la determinación del umbral que separa los niveles de intensidad en cada pixel de la imagen. (T. Acharya; A. K. Ray, 2005)

- Segmentación basada en el área o por región: Dentro de este tipo de segmentación, se encuentran los métodos de crecimiento de regiones y división-fusión, donde se busca dividir la imagen en particiones que comparten ciertas características comunes. El método de crecimiento de regiones se basa en los criterios de conectividad y similitud de puntos. Y el método por división y fusión se basa en la unión de regiones, normalmente las dos fases del algoritmo se efectúan en este orden, primero una división en regiones para posteriormente unir aquellas regiones vecinas similares. (T. Acharya; A. K. Ray, 2005)
- Segmentación mediante detección de contorno: Para realizar esta segmentación, es necesario entender que un contorno es un conjunto conectado de pixeles que tienen el mismo nivel de intensidad. Los contornos de una imagen se pueden clasificar en dos categorías, contornos definidos por la intensidad de los pixeles y contornos definidos por la textura de una región. La detección de contornos se realiza mediante búsqueda de discontinuidades en los niveles de grises de la imagen, es decir, de pixeles en los cuales el nivel de grises varié abruptamente respecto al de sus vecinos. (T. Acharya; A. K. Ray, 2005)
- Segmentación basada en principios o modelos físicos: esta segmentación está basada en principios o modelos ópticos y físicos, se fundamenta en el estudio del proceso de generación de la imagen misma, considerando la forma en que se da la reflexión de la luz sobre la superficie de interés de acuerdo a su material, de las características de la cámara u óptica empleada y la forma en que se ubican las luces y se proyectan las sombras. Este método es similar a los otros métodos de segmentación, en cuanto a la implementación de algoritmos, sólo que difiere de ellos por el hecho de usar explícitamente los modelos de reflexión de las

superficies como herramientas para la segmentación de las imágenes. (R. C. Gonzáles; P. Wintz, 1996)

- Transformación de Watershed: La transformación Watershed es una aproximación matemática por morfología. Considera la magnitud del gradiente de una imagen, interpretándola de forma similar a una superficie de contornos topográfica. El nivel de gris de los píxeles indicaría entonces la altitud y los píxeles conexos que tengan mayor magnitud de gradiente corresponden a las líneas que delimitan los contornos o Watershed, que representan los límites de la región. (R. C. Gonzáles; P. Wintz, 1996)

3.3 Componentes Electrónicos

Una parte importante en el proceso de diseño es la elección de los componentes electrónicos, los cuales estarán encargados de la mayor parte del proceso de la experimentación. Entre ellos se encuentran tarjetas de desarrollo, ordenadores de placa reducida, sensores y motores.

3.3.1 *Arduino*

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el proceso de programación en proyectos de electrónica multidisciplinarios. El hardware consiste en una placa con un microcontrolador Atmel y puertos de entrada/salida. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa.

Para este trabajo fue elegido el Arduino Uno, dado que posee las especificaciones requeridas para este proyecto, tomando en cuenta la cantidad de pines de entrada/salida, la compatibilidad con los otros componentes, y el tamaño.

Microcontrolador	Armega328
Voltaje de operación	5V
Voltaje de entrada(Recomendado)	7 – 12V
Voltaje de entrada(Limite)	6 – 20V
Pines de entrada-salida digital	14(6 pueden usarse como salidas de PWM)
Pines de entrada analógica	6
Corriente continua por pin IO	40mA
Corriente continua en el pin 3.3V	50mA
Memoria Flash	32 KB(0.5 KB ocupados por el bootloader)
SRAM	2KB
EEPROM	1KB
Frecuencia de reloj	16MHz

Tabla 1. Características del Arduino Uno

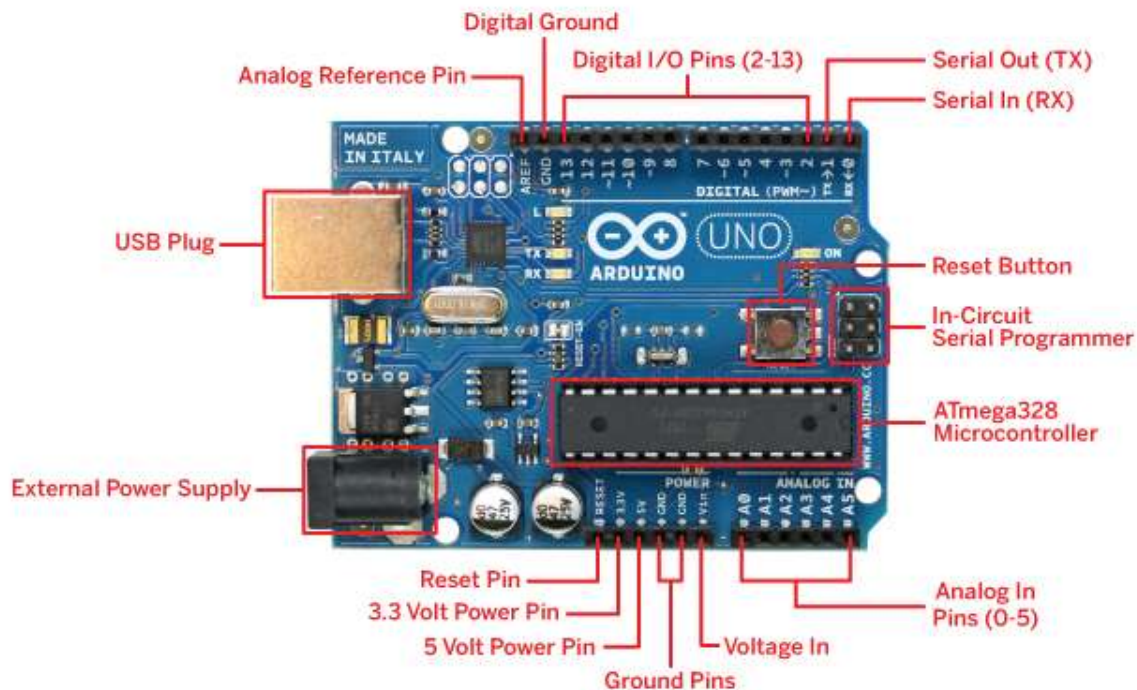


Figura 4. Componentes del Arduino Uno

3.3.2 Raspberry Pi B

La Raspberry Pi es un ordenador de placa reducida de bajo costo, diseñada para estimular la enseñanza de ciencias de la computación en las escuelas. El diseño incluye un sistema integrado Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM. El diseño no incluye un disco duro ni una unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente.

Para este trabajo se utilizará la Raspberry Pi modelo B, ya que posee los mínimos recursos para el correcto funcionamiento del dispositivo y software.

	Modelo A	Modelo B	Modelo B+
SoC	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB)		
CPU	ARM 1176JZF-S a 700MHz (familia ARM11)		
Juego de instrucciones	RISC de 32 bits		
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, MPEG-2 y VC1.		
Memoria (SDRA)	256 MB (compartidos con la GPU)	512 MB (compartidos con la GPU)	
Puertos USB 2.0	1	2 (vía hud USB integrado)	4
Entradas de video	Conector MIPI CSI que permite instalar un módulo de cámara desarrollado por la RPF		
Salida de video	Conector RCA (PAL y NTSC), HDMI (rev1.4 y 1.4),interfaz DSI para panel LCD		
Salida de audio	Conector de 3.5 mm, HDMI		
Almacenamiento integrado	SD/MMC/ranura para SDIO		MicroSD

Conectividad de red	Ninguna	10/100 Rthernet (RJ-45) via hub USB	
Periféricos de bajo nivel	8 x GPIO, SPI, I2C, UART		
Reloj en tiempo real	Ninguno		
Consumo energético	500mA, (2.5W)	700mA,(3.5W)	600mA,(3.0W)
Fuentes de alimentación	5V vía Micro USB o GPIO header		
Dimenciones	85.60mm x 53.98mm		
Sistemas operativos soportados	GNU/Linux: Debian(raspbian),Fedora(Pidora), Arch Linux(Arch Linux ARM), Slackware Linux. RISC OS		

Tabla 2. Características técnicas de la Raspberry Pi

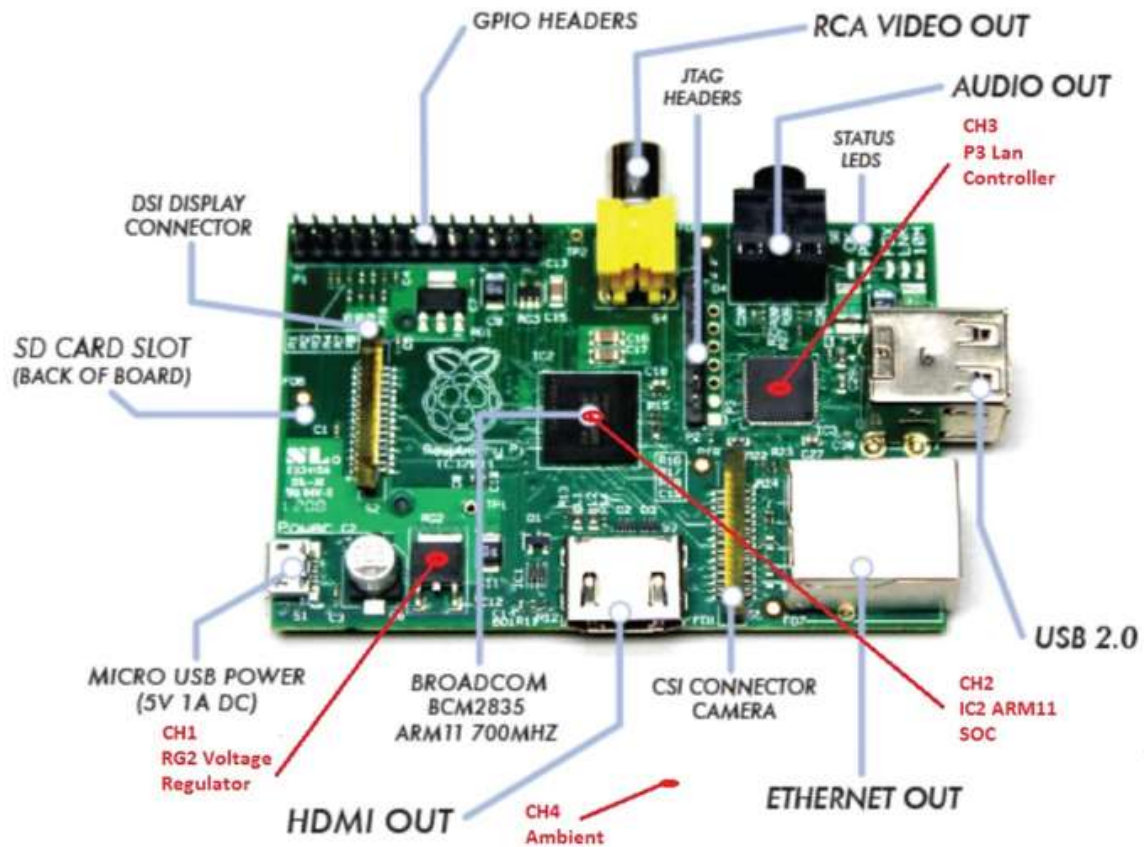


Figura 5. Raspberry Pi modelo B

3.3.3 Webcam Philips SPC1300NC

Una webcam (cámara web o de red) es una pequeña cámara digital conectada a un ordenador, la cual puede capturar imágenes y transmitir las a través de algún tipo de conexión, en este caso se utilizará la webcam Philips SPC1300NC, que proporciona unas buenas características de desempeño para la experimentación que se desea realizar.

Sensor	1.3MP CMOS
Real video resolution(Max)	1.3MP
Real still image resolution(Max)	1.3MP
Interpolated still image resolution(Max)	5MP
Frames per second	90 fps
F-number	2.8
Lens view angle	80 degrees
Minimum subject illumination	<5 lux
Data format	1420, IYUV
Interfacing	UVC
USB cable length	2.1 m
Microphone	2 digital built-in omni directional microphones
Power	Supplied via USB cable
Voltage requirements	5V, 0.5 ^a

Tabla 3. Características webcam Philips SPC1300NC



Figura 6. Webcam Philips SPC1300NC

3.3.4 Servomotor Dongbu Robot Herkulex DRS-0101

Un servomotor es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Otra característica del servomotor es que puede ser controlada su velocidad para lograr una correcta posición, y la cantidad de fuerza que maneja puede ser superior a la de los motores de corriente continua normalmente, estas características son utilizadas para controlar diversos componentes, por ejemplo una pieza de robótica.

Para el caso de este trabajo, es necesario controlar la posición en grados de un disparador de proyectiles, dado el limitado espacio y la fuerza que se necesita para

manipular el disparador, fue elegido el servomotor dongbu robot herkulex DRS-0101, el cual cumple a cabalidad las necesidades propuestas por el problema.

Input voltaje	7.4V DC
Stall torque	12Kgf.cm @ 7.4V
Maximim speed	0.166s/60° @ 7.4V
Operating angle	320°

Tabla 4. Características servomotor dongbu robot herkulex DRS-0101



Figura 7. Servomotor dongbu robot herkulex DRS-0101

4. DESARROLLO DEL PROYECTO

Este proyecto tiene como objetivo el diseño y la construcción de un dispositivo de experimentación para laboratorio de cinemática, el cual se apoye en un sistema de embebido con sistema operativo. Para lograr este objetivo se requiere el diseño y elaboración de una estructura que posea las condiciones apropiadas para la experimentación; tales como el manejo de posición, fuerza de disparo y recarga automática de proyectiles. A su vez, requiere el diseño de un software que permita la adquisición de datos por medio de sensores y su visualización en una pantalla.

En primer lugar, se diseñó la estructura utilizando el programa Solidworks 2014, donde esta estructura está diseñada para que limite las condiciones físicas, para que la toma de datos de un cuerpo en movimiento sea lo más clara posible. Con el fin de obtener datos de movimiento, esta estructura cuenta con un sistema de auto recarga, un disparador, una pared que limita la distancia recorrida por el objeto, además de una superficie recubierta con cera para reducir la fricción.

4.1 Descripción de la Estructura

El diseño y construcción de la estructura debe estar sujeto a la resolución de la cámara, con el fin de abarcar el espacio suficiente para que el proyectil se desplace y pueda ser capturado por la misma, en este caso la cámara (Philips SPC1300NC) que se utiliza tiene una relación de 4:3, esto permite que la máxima distancia en la que el proyectil es identificado por la cámara sea de 50 cm, utilizando esta medida, se dio una relación de tamaño para que la ventana donde se toman los datos tenga una medida de 61 cm de base por 45 cm de altura.



Figura 8. Ventada de captura de información.

El diseño de la estructura posee dimensiones de 80 cm de base, 61 cm de altura y 21.5 cm de profundidad, teniendo en cuenta el espacio necesario para el funcionamiento de los diferentes sistemas que posee la estructura, estos son: el sistema de disparo, el sistema de ajuste de ángulo, el sistema de ajuste de altura y el sistema de recarga del proyectil.



Figura 9. Foto frontal de la estructura completa de experimentación.



Figura 10. Dirección de movimiento de los sistemas que posee la estructura de experimentación.

La disposición de los elementos fue pensada para lograr una buena captura de la información respecto al movimiento que genera el proyectil al ser disparado. Por tal motivo, el disparador se encuentra a la izquierda de la ventana de captura de información, y el cargador de proyectiles se encuentra dispuesto en el extremo izquierdo del disparador, lo cual permite un máximo en el desempeño de la estructura a la hora de la captura de información.

La ventana de captura de imagen tiene una profundidad de 12 cm, lo cual ayuda en la limitación de corrientes de aire externas, que puedan afectar el desplazamiento del proyectil. El fondo de la ventana posee un panel de color blanco mate, que limita el brillo de la iluminación que recibe la estructura en la toma de datos, también la ventana está protegida por un acrílico transparente que evita que el proyectil salga de la estructura si el disparo tiene demasiada potencia, o es disparado con un error de posición y este pudiera chocar contra las paredes. Agregado a esto, la superficie de la ventana de captura posee

una capa de cera, que proporciona un mínimo de fricción al contacto con el proyectil. Al eliminar la mayor cantidad de factores que afecten la toma de datos o la trayectoria del proyectil, se logrará una mayor fidelidad para obtener valores del movimiento y la trayectoria del objeto.

En la parte frontal de la estructura se encuentra una canaleta con una inclinación de 10° , que tiene como función el transporte del proyectil al finalizar el disparo, y lograr posicionarlo en el sistema de recarga. Los diferentes sistemas que posee la estructura son controlados desde una consola que tiene tres perillas, para controlar la altura del disparador, el ajuste en el ángulo y la potencia del disparo, y un pulsador que se encarga de activar la recarga del proyectil.

El proyectil que se manipula a lo largo de la experimentación, es una esfera de madera que posee un diámetro de 2.5 cm, y su exterior posee una tonalidad rojiza, este color facilita el sistema de captura de imagen para lograr identificar el objeto al cual se le hará el procesamiento de imagen.

4.1.1 Disparador

El disparador consta de tres piezas, la primera es un codo plástico de una pulgada que está posicionado con uno de sus extremos de manera vertical, lo cual permite la entrada del proyectil al disparador. El otro extremo de esta pieza está dispuesto de manera horizontal, esta pieza esta fija a la plataforma que se utiliza para ajustar la altura del disparador.

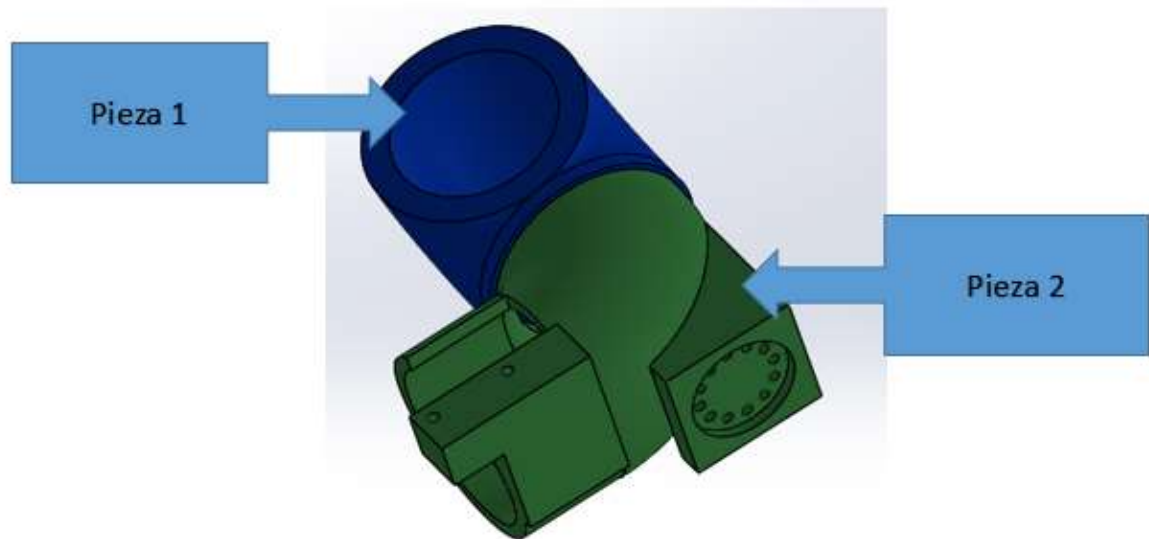


Figura 11. Pieza 1 y pieza 2 del sistema de disparo.

La segunda pieza consta de un segundo codo de una pulgada que está unido al extremo horizontal libre de la primera pieza, lo cual da la posibilidad de rotar la segunda pieza en su propio eje. El extremo libre de la segunda pieza, en la parte superior, tiene una hendidura de 2.3 cm y una profundidad de 2.5 cm, esta tiene la función de proporcionar el espacio suficiente para la rotación de la tercera pieza. La segunda pieza en el costado horizontal, opuesto a la primera pieza, posee una unión para anclar el sistema de ajuste de ángulo, y en el extremo libre existe una unión para aferrar la tercera pieza.

La tercera pieza es un micro motor dc con una relación de 5:1 y genera 6000 rpm, este se encuentra sujeto a la parte libre de la segunda pieza, con el fin de mover una rueda plástica de 2.5 cm de diámetro y un grosor de 2 cm, la cual funciona como un propulsor al proyectil y se encuentra en la parte superior de la hendidura, con la finalidad de poder disparar el proyectil, la rueda plástica tiene una separación de la parte interna de la segunda pieza de 2.5 cm, es decir, el diámetro del proyectil, esto para lograr conseguir el mejor agarre de la rueda plástica al proyectil para asegurar un correcto disparo.

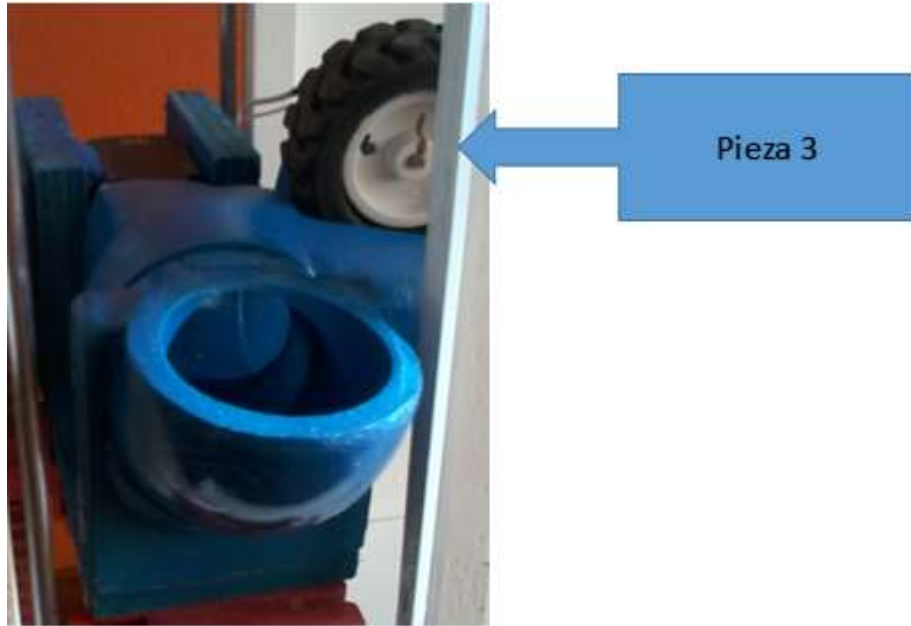


Figura 12. Sistema de disparo

4.1.2 Ajuste de Ángulo

El sistema de ajuste de ángulo está conformado por tres piezas, la primera pieza es el sistema de disparo, el cual se explicó anteriormente, este cuenta con una unión en un lateral que le permite anclarse con la segunda pieza.

La segunda pieza es un servo motor (Herkulex DRS 0101) que se encuentra fijo a la plataforma de ajuste de altura de manera paralela horizontal al extremo de alimentación de proyectiles de la primera pieza, esto con el fin de que al activar este servomotor, el sistema de disparo logre modificar el ángulo de inclinación en el cual se encuentre respecto al eje vertical de la salida del proyectil.

Como tercera pieza se tiene la plataforma que tiene unas dimensiones de 4.5 cm de base, 15.5 cm de largo y 6 cm de altura, a la cual se encuentra fija la pieza uno, en el extremo de entrada de proyectiles al disparador, y la segunda pieza, el servo motor en el

lado opuesto. El tamaño de la tercera pieza da a la estructura de disparo la posibilidad de ajustar el ángulo entre los -30° hasta los 65° .

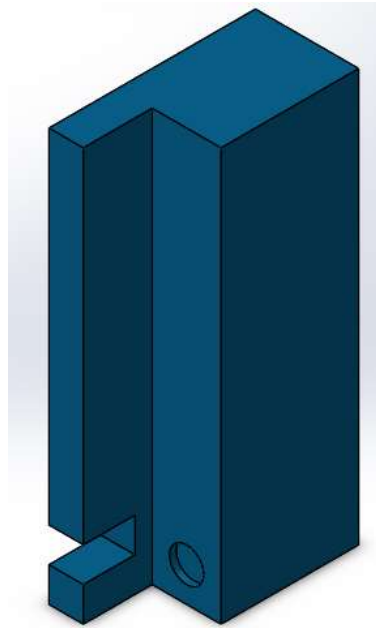


Figura 1. Unión entre la plataforma y el tornillo sin fin.

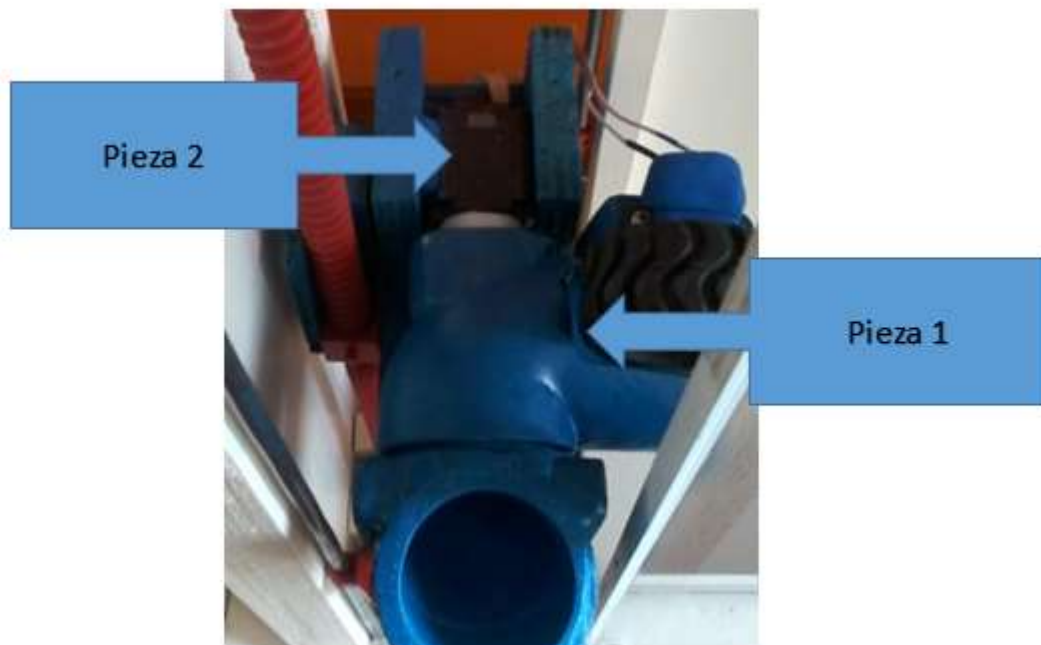


Figura 14. Parte superior del sistema de ajuste de ángulo.

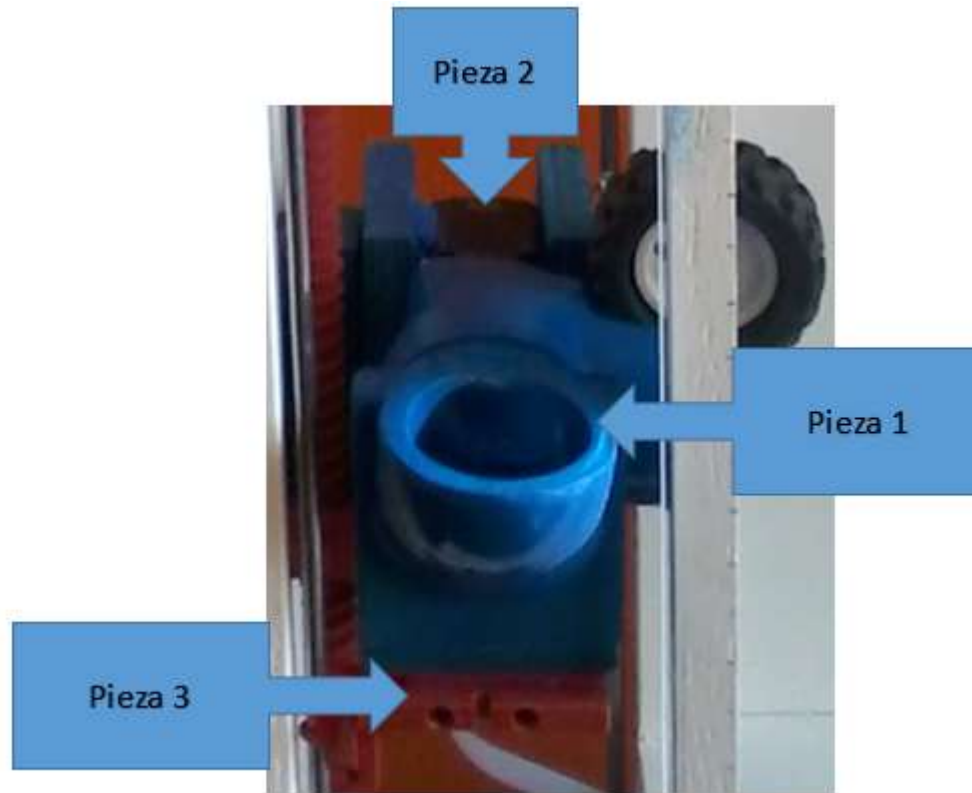


Figura 15. Parte frontal del sistema de ajuste de ángulo.

4.1.3 Ajuste de Altura

El ajuste de altura para la plataforma de disparo cuenta con nueve diferentes piezas de la empresa Fischertechnik, las cuales permiten que la estructura de disparo para el proyectil logre subir o bajar dependiendo de la necesidad de la experimentación, con la posibilidad de desplazarse dentro de 38 cm.

La primera pieza es un motor con encoder (Figura 23) que maneja una relación de 25:1, el cual se encuentra recostado en la parte inferior de la estructura, este está conectado por una pieza cilíndrica (Figura 24) la cual une el motor con una corona dentada (Figura 25) (Figura 16), para evitar la presión sobre las piezas de unión, se coloca una pieza cubica (Figura 26) que se utiliza como refuerzo.

Se tiene una segunda pieza de corona dentada conectada a la primera y unida en la parte superior con otra pieza cubica (Figura 27), esto con el fin de transmitir el movimiento del motor a la parte superior de la segunda corona dentada. (Figura 17)



Figura 16. Corona plástica y unión

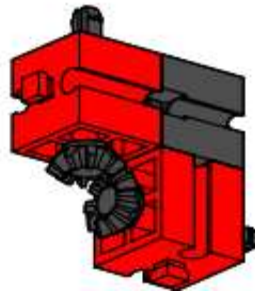


Figura 17. Estructura de coronas para movimiento del motor y el sistema de ajuste de altura

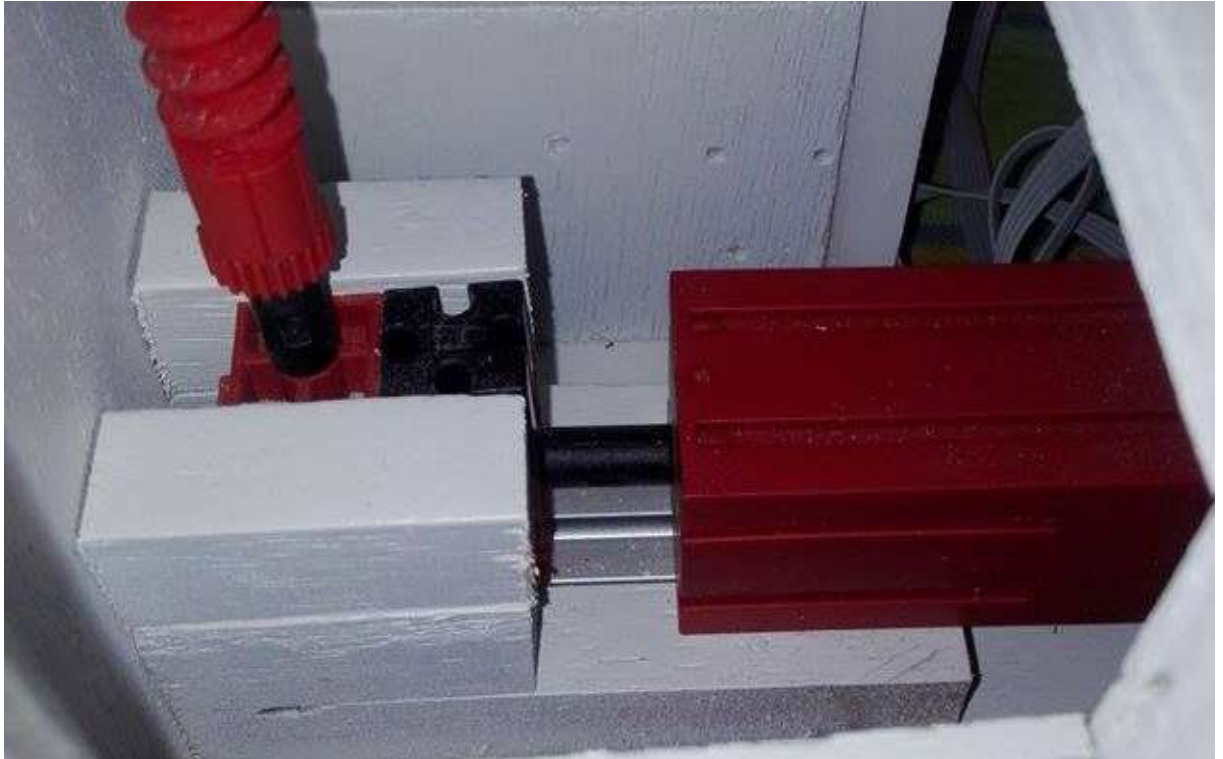


Figura 18. Estructura mecánica para el ajuste de altura de la plataforma de disparo

Se tiene un tornillo sin fin construido con 12 piezas en forma de tornillo huecas (Figura 28), y en su centro se unen utilizando una varilla de 0.3 cm de diámetro y 51 cm de largo, en la parte inferior tiene un tope de protección (Figura 29) (Figura 30) para asegurar las piezas a la unión de la segunda corona dentada, y en la parte superior dos toques (Figura 31) se seguridad.

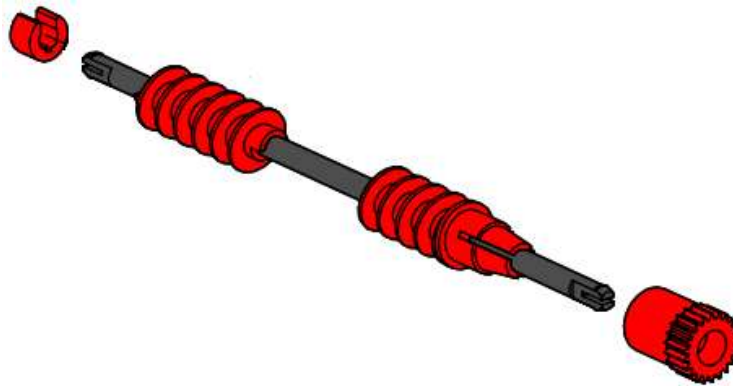


Figura 19. Unión de las piezas para el tornillo sin fin.

Como el motor se encuentra anclado en la parte interior, conectado a un sistema que transmite el movimiento de este hacia un tornillo sin fin, al entrar en funcionamiento se coloca una pieza cubica (Figura 32) en este sistema, el cual se puede desplazar libremente de manera vertical sobre el tornillo sin fin (Figura 20), esta pieza se encuentra sujeta a la plataforma de disparo, para lograr equilibrio al desplazar esta pieza junto a la plataforma, se coloca un seguro en la unión, y en la plataforma se agregaron dos uniones (Figura 33) (Figura 34) sujetas a unas varillas para facilitar (Figura 21) el desplazamiento y que la plataforma no pierda su punto de inercia.

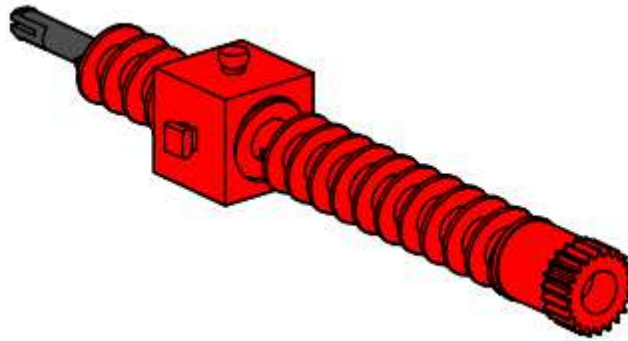


Figura 20. Tornillo sin fin armado con la unión plástica para el ajuste de altura.



Figura 21. Pieza de la unión para el soporte de la plataforma.

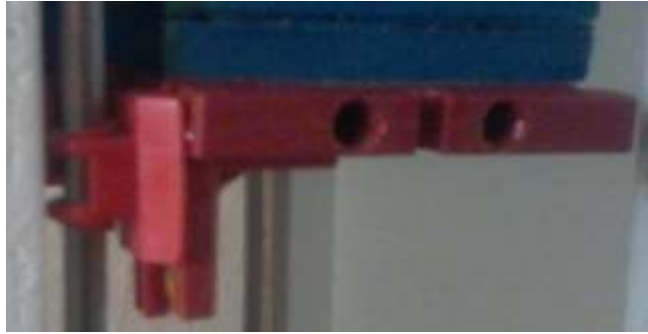


Figura 22. Soportes de la plataforma para facilitar el desplazamiento.

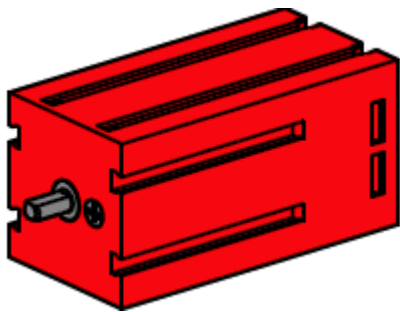


Figura 23. Motor con encoder.

Referencia 135 485



Figura 24. Unión plástica.

Referencia 35 073



Figura 25. Corona dentada plástica.

Referencia 35 061

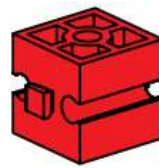


Figura 26. Unión plástica.

Referencia 32 064

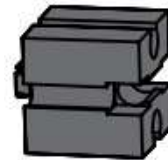


Figura 27. Unión plástica.

Referencia 32 881

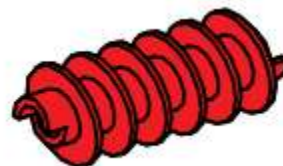


Figura 28. Unión plástica con relieve torneado. Referencia 37 926

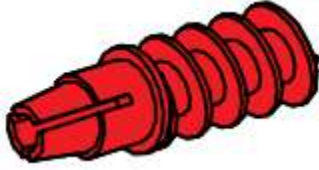


Figura 29. Unión plástica con relieve torneado y unión para broche.
Referencia 37 858

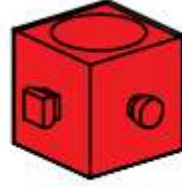


Figura 32. Unión plástica con centro torneado. Referencia 37 925



Figura 30. Broche plástico.
Referencia 31 915



Figura 33. Unión plástica.
Referencia 31 426



Figura 31. Broche plástico.
Referencia 37 679



Figura 34. Unión plástica.
Referencia 38 423

4.1.4 Auto Recarga

El sistema de recarga cuenta con catorce diferentes piezas de la empresa Fischertechnik, las cuales le permiten al proyectil trasladarse desde una posición hasta el disparador. Este sistema está conformado por tres estructuras, la primera se encarga de mover una cadena transportadora, la segunda es un eje de soporte para la cadena de transporte, y la tercera es una plataforma para transportar el proyectil hasta el disparador.

La primera estructura tiene un motor (figura 38) que maneja una relación 25:1, está unido a una rueda dentada (Figura 39) por medio de una barra plástica (Figura 40), una unión cilíndrica (Figura 24), un soporte plástico (Figura 41) y un cubo (Figura 26). En la parte superior cuenta con dos piezas rectangulares (Figura 42) que funcionan de soporte para la rueda dentada. El motor y soportes están anclados en una caja de madera la cual permite el correcto funcionamiento y no permiten que se descuadre la posición en la cual se está.

La segunda estructura es un eje de soporte para la cadena de transporte, está construida con dos soportes rectangulares (Figura 42) que se encargan de apoyar una rueda dentada (Figura 39), la cual tiene una barra plástica (Figura 40) como centro, un soporte plástico y en la contra parte está bloqueada con un tope de seguridad (Figura 31).

La tercera parte del sistema de recarga consiste en dos pequeñas plataformas sujetas a una cadena plástica armada con 275 broches (Figura 45) y tiene un largo de 110 cm (Figura 37) que sirve para transportar el proyectil. Estas plataformas están construidas utilizando una pieza plástica de soporte (Figura 43) que está anclada a la cadena, a esta pieza está unida una pieza rectangular que tiene una de sus caras inclinada (Figura 44), del lado inclinado de la pieza están conectados dos cubos (Figura 27) que se encargaran de sostener el proyectil el tiempo en que es transportado a el disparador.

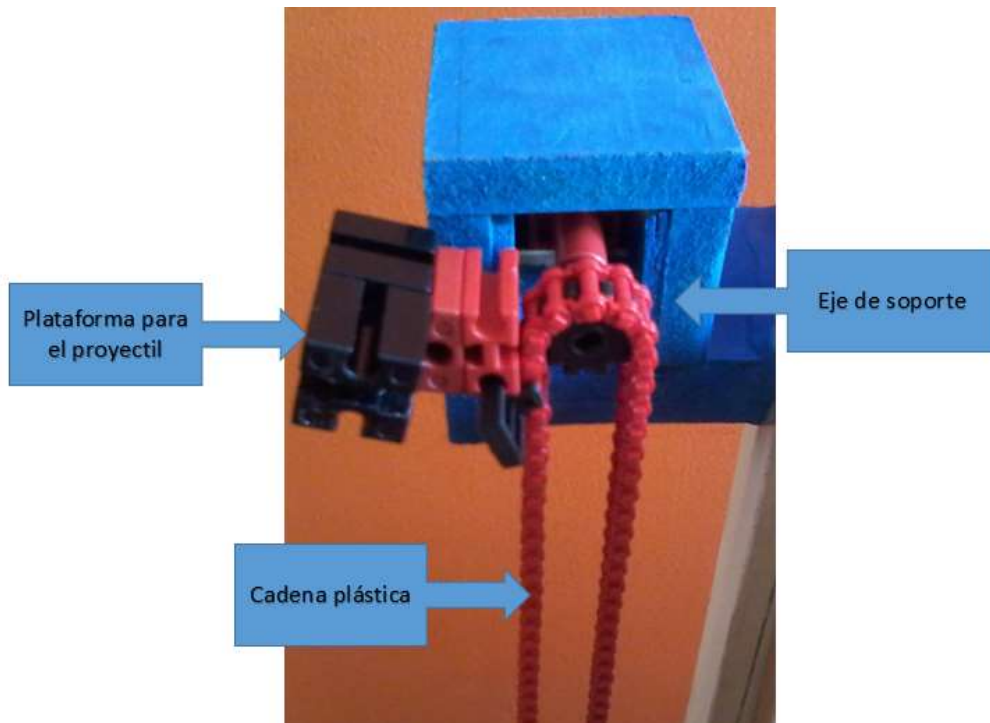


Figura 35. Eje superior del sistema de recarga conectado a la cadena plástica y plataforma para el proyectil.

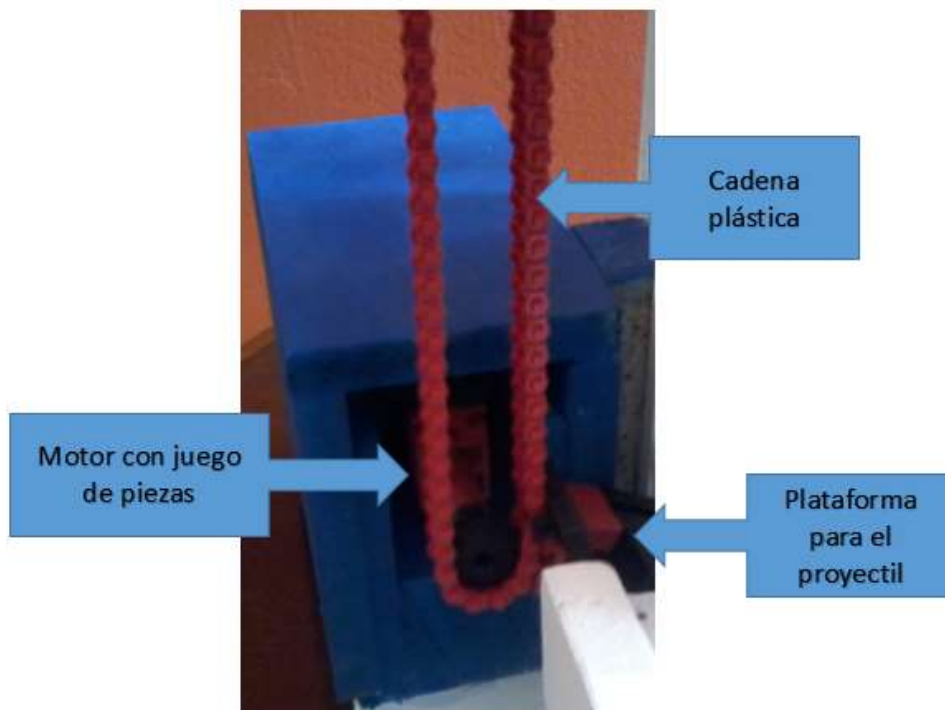


Figura 36. Eje inferior conectado al motor, junto con la cadena plástica y plataforma para el proyectil.

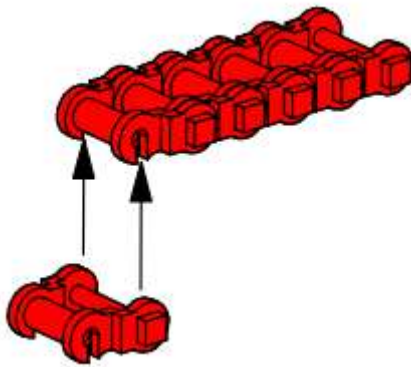


Figura 37. Armado de cadena plástica.

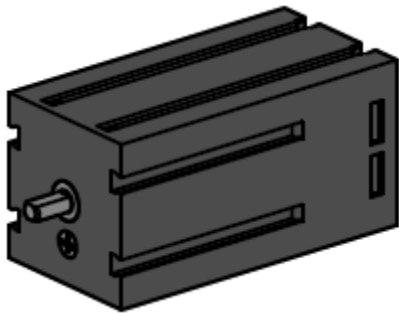


Figura 38. Motor.
Referencia 135 485



Figura 39. Unión plástica con corona dentada.
Referencia 35 945



Figura 40. Unión plástica.
Referencia 35 065



Figura 41. Unión plástica.
Referencia 31 983

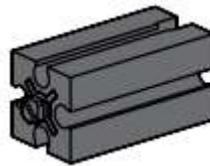


Figura 42. Unión plástica.
Referencia 32 879



Figura 43. Unión plástica para cadena.
Referencia 31 762



Figura 44. Unión plástica con inclinación de 7.5°. Referencia 32 071



Figura 45. Pieza de armado de cadena plástica. Referencia 36 248

4.1.5 Controles de Estructura

Los diferentes sistemas que componen el funcionamiento de la estructura pueden ser vistos como variables controladas por el usuario para lograr una experimentación de laboratorio para física cinemática, para ello es necesario tener una consola de controles, para manipular estas variables.

Las variables a controlar son el ángulo en el cual se tenga la salida del disparador, la altura en la cual se quiere tener el disparador y la potencia del disparo con la cual va a ser lanzado el proyectil a la hora de la experimentación.

Estas variables junto al sistema de recarga del proyectil son manipuladas por tres perillas y un pulsador, estos dispositivos cumplen la función de ajustar las variables de control y el pulsador activar el sistema de recarga del proyectil.



Figura 46. Controles de los sistemas de la estructura para la experimentación.

Los datos entregados por estos cuatro elementos son leídos por los puertos analógicos de Arduino que son valores entre 0V hasta los 5V he interpretados como 0 hasta 1023 (10 bits), estos datos son procesados y por medio de los puertos digitales de Arduino entregados a los sistemas de control respectivamente.

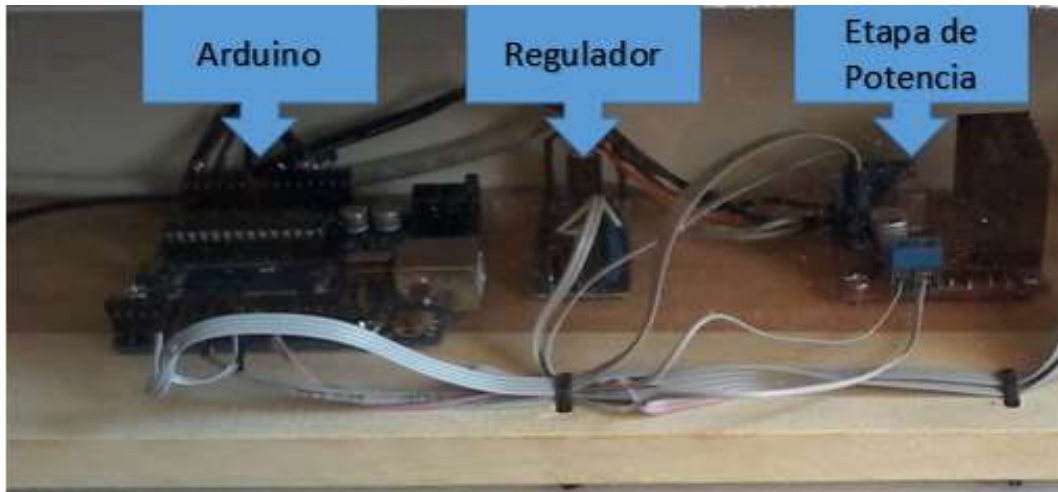


Figura 47. Componentes electrónicos para el control de la estructura.

El ajuste de ángulo es controlado por el servo motor (Herkulex DRS 0101), para lo cual el Arduino procesa los datos entregados por la primera perilla, estos datos son mapeados para poder leerlos como valores entre de -30 hasta 65, que corresponde a los grados en los cuales el disparador se puede posicionar.

El sistema de ajuste de altura es controlado por el motor con encoder de Fischertechnik, para ajustar la posición en la que la plataforma de disparo se va a posicionar se utiliza la segunda perilla, los datos entregados por esta son mapeados en el Arduino para tener tres valores de interés, los cuales son el 0, 1 y 2. El ajuste de altura maneja tres posiciones, la primera hace que descienda la plataforma de disparo, la segunda apaga el motor para dejar la estructura en una posición fija, y la tercera para que la plataforma ascienda. La alimentación del motor está dada por una etapa de potencia, la cual esta manejada por un pwm (modulación por ancho de pulso) entregado por la placa de Arduino, esto es necesario dado que los pines de salida del Arduino no son capaces de generar la corriente necesaria para el correcto funcionamiento de este motor.

La potencia de disparo del proyectil está controlada por la velocidad de giro de un micro motor, la velocidad de este es manipulable por medio de la tercera perilla. Esta cuenta con 7 marcas las cuales indican la potencia de disparo que maneja el disparador, los datos que son entregados por esta perilla son mapeados por Arduino para tener valores de 0 a 255 que hacen referencia al voltaje de salida en los pines digitales, que manejan entre 0 a 5 voltios. Los valores digitales de salida que manejan la velocidad del micro motor del disparador son controlados por un pwm , dado que la velocidad es ajustada mediante la alimentación de voltaje a este, para lo cual la salida digital de la placa de Arduino está conectada a una etapa de potencia.

El pin digital a utilizar maneja valores que están entre los 10 y los 70, tomando como 10 el valor mínimo en el cual el disparador es capaz de lanzar el proyectil, y 70 la máxima capacidad de disparo, esto da que el micro motor del disparador solo esté funcionando hasta un 27.45%, esto es porque a un mayor nivel o voltaje de salida, el proyectil alcanza una velocidad y distancia mayor a la que la estructura puede manejar.

El sistema de recarga de proyectiles es activado por medio de un pulsado normalmente abierto, que al ser pulsado genera un dato que es leído por un pin analógico de la placa de Arduino, esto genera un dato de salida en un pin digital controlado por un pwm, para la potencia del motor de Fischertechnik y la duración de la activación de este es de 5,5 segundos, el tiempo necesario para recoger el proyectil y desplazarlo hasta el disparador.

El controlador tiene unas dimensiones de 17.5 cm de largo 9.5 cm de ancho y 8.5 cm de alto, las perillas se encuentran centradas en la parte frontal y el pulsador posicionado debajo de la perilla central, cada perilla cuenta con unas marcas que indican en funcionamiento de cada una. El cableado y componentes electrónicos están localizados en la parte trasera de la estructura final, con el fin de proporcionar un correcto cableado y no afectar el funcionamiento de los mismos a la hora de la experimentación.

4.2 Descripción del Software

Para el diseño y aplicación del software se trabajó utilizando como herramienta el programa Matlab para hacer cross compiler (compilador cruzado) para la Raspberry Pi, dado que Matlab permite la programación en Simulink (entorno de programación visual), utilizando el soporte para la Raspberry Pi. Este paquete de soporte permite al usuario programar y utilizar los puertos externos, cámara y otros componentes, que posee o se le pueden adicionar a este ordenador de placa reducida para facilitar el correcto proceso de captura, procesamiento y posteriormente visualización de los resultados.

A continuación se presenta el proceso llevado a cabo para la captura y procesamiento de imagen.

4.2.1 Captura de Imagen

Prueba 1. Captura directa.

Se utilizó el bloque 'V4L2 Video Captura' que se encarga de captura de imagen desde una webcam conectada a la Raspberry pi, y el bloque 'SDL Video Display' para la visualización de video, con el fin de identificar los parámetros a manejar que brindan estos elementos. De estos bloques de captura y visualización se logró identificar el tamaño de la imagen a capturar por el bloque de webcam, el tipo de dato que entregará este bloque en términos de RGB o Ycbcr, y el tiempo de muestreo al que va a funcionar.

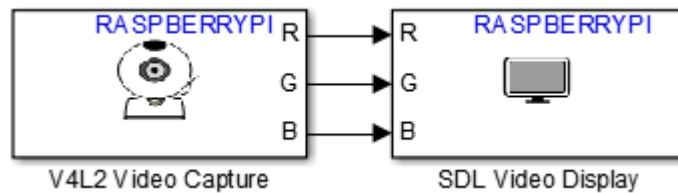


Figura 48. Conexión de bloques de captura de imagen por webcam y salida de imagen por monitor.

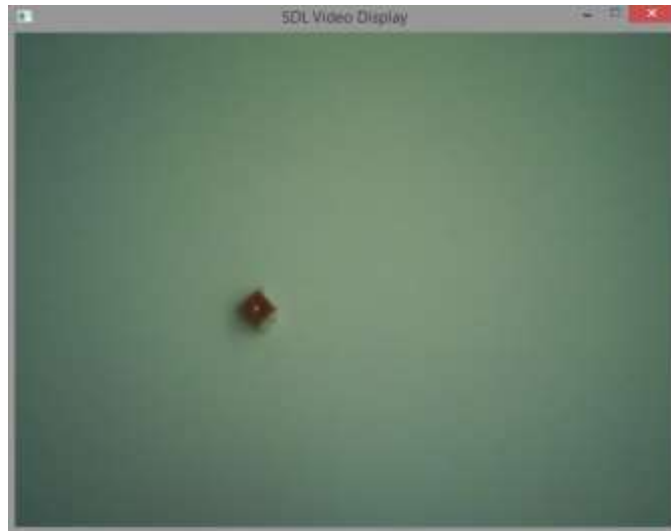


Figura 49. Imagen a la salida del bloque de video.

Prueba 2. Escala de grises.

Se agregaron los bloques de 'Color Space Conversion' para efectuar la conversión de datos de color a escalas de grises, y 'Demosaic' para volver el dato de escala de grises a tonos de color. Esto con el fin de incorporar bloques de otros paquetes a la simulación, dado que no todas las aplicaciones que brinda simulink pueden ser ejecutadas desde la Raspberry pi. Además de que en el tratamiento de señales es muy importante el poder convertir una imagen que este en formato de colores a una de escala de grises.

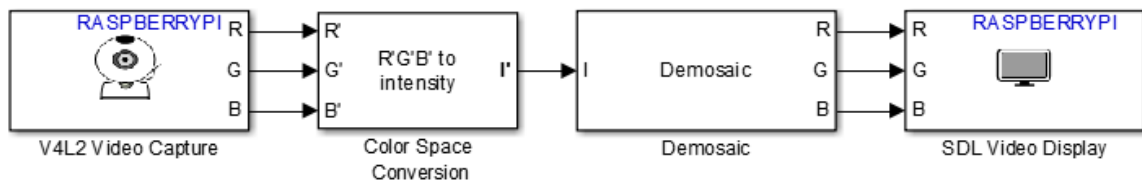


Figura 50. Conexión de bloques de captura de imagen por webcam y salida de imagen por monitor, y bloques de conversión a grises y reconversión a RGB.



Figura 51. Imagen a la salida del bloque de video.

Prueba 3. Utilizando funciones de Matlab.

Se utilizó el bloque 'Matlab Function', el cual permite utilizar código en la simulación, esto con el fin de probar que es posible ejecutar sin problema códigos de Matlab en la Raspberry pi, lo cual genera una correcta simulación, pero afecta en la velocidad de procesamiento y adquisición de datos adquiridos desde la webcam, además de que la

gran mayoría de funciones que posee Matlab no pueden ser llamadas por código, esto crea una dificultad a la hora de programar.

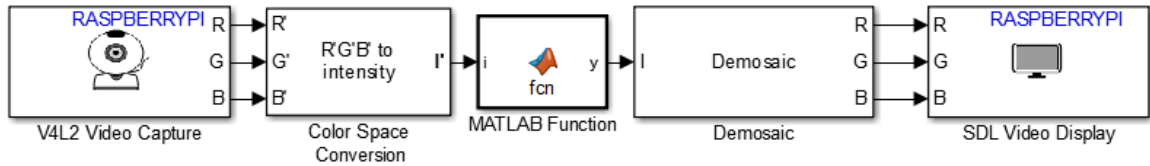


Figura 52. Conexión de bloques de captura de imagen y salida, bloques de conversión y reconversión, y prueba de Matlab Function.

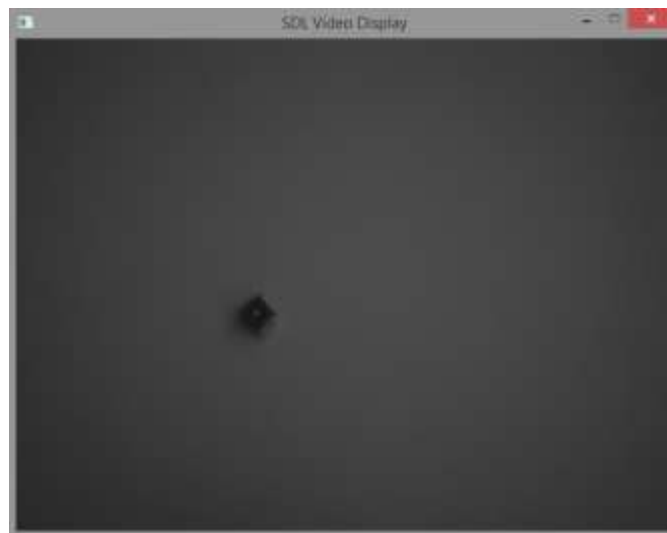


Figura 53. Imagen a la salida del bloque de video.

Prueba 4. Identificación de rectángulos.

Se emplearon los bloques 'Edge Detection' y 'Blob Analysis' en la detección de rectángulos, específicamente para lograr comprender el funcionamiento y parámetros de estos bloques, los cuales se encargan de buscar o identificar colores y formas en archivos de imagen, y posteriormente utilizar el bloque 'Draw Shaper' para generar bordes en el objeto detectado en la imagen.

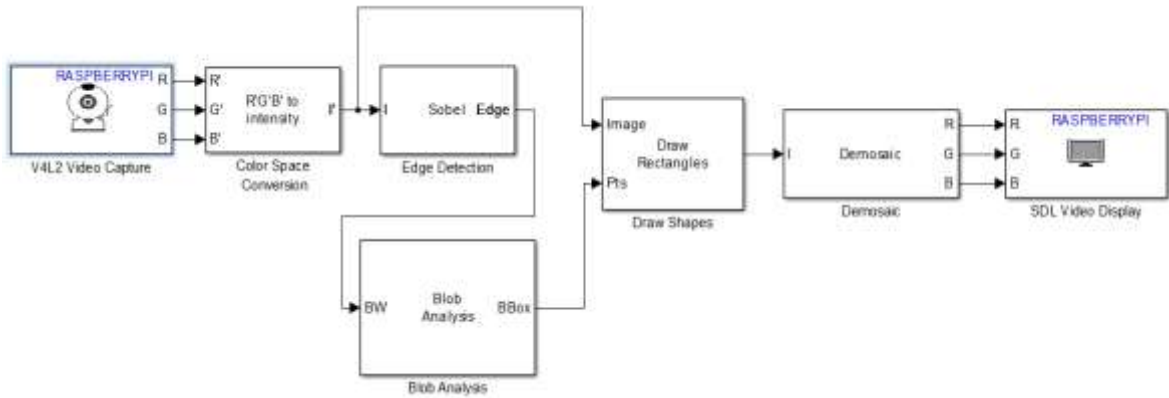


Figura 54. Proceso de captura y muestra de imágenes con reconocimiento de bordes rectangulares.

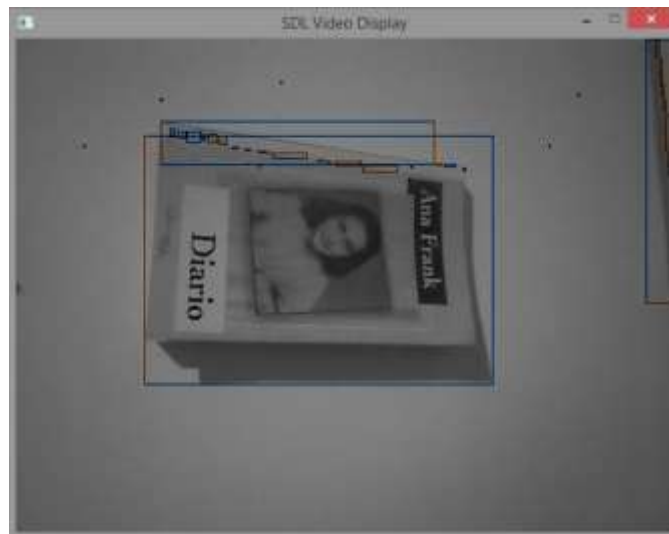


Figura 55. Identificación de rectángulos con un libro que posee esta forma.

Prueba 5. Identificación de círculos.

Dado que el proyecto requiere proyectiles redondos, es necesario identificar la forma circular, para lo cual apoyados en la simulación anterior, donde se identificaron rectángulos, se empleó una simulación similar para reconocer la forma de interés.

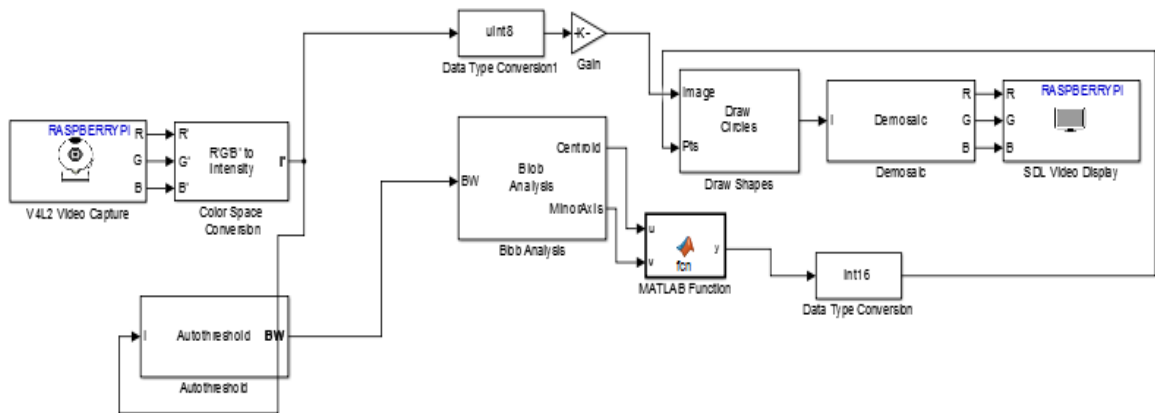


Figura 56. Proceso de captura y muestra de imágenes con reconocimiento de bordes circulares.

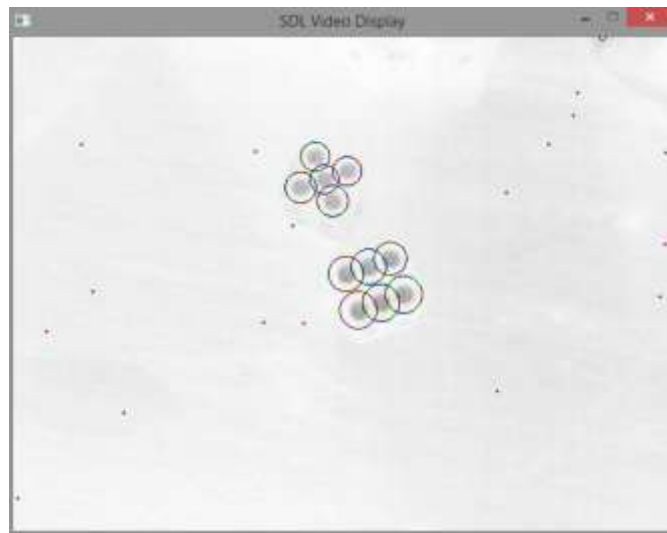


Figura 57. Identificación de círculos utilizando dos datos.

Prueba 6. Identificación de color.

Después de trabajar la simulación utilizando los bloques disponibles en simulink para el procesamiento de imágenes, se utilizó el bloque de 'Matlab Function' para hacer procesamiento de imagen por código.

Dado que la gran mayoría de funciones que posee Matlab por código no pueden ser utilizadas en la simulación desde la Raspberry pi, fue necesario generar un tratamiento de imagen más sencillo.

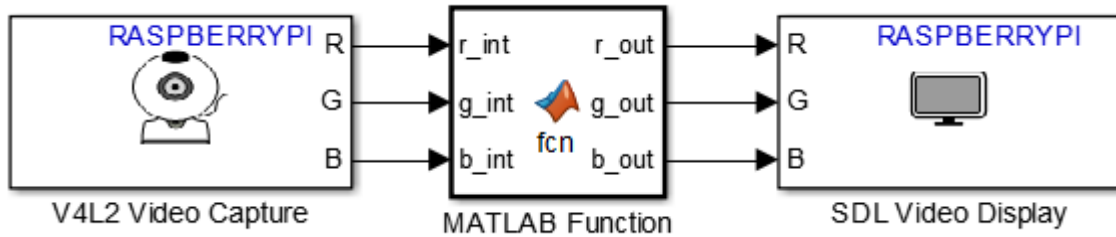


Figura 58. Tratamiento de imagen con el bloque Matlab Function.

El código que se encuentra en el bloque ‘Matlab Function’ necesita como entrada tres matrices de colores en RGB (rojo, verde, azul), los cuales son suministrados por el bloque ‘V4L2 Video Capture’. Estas tres matrices son guardadas dentro de la función para crear una estructura o imagen nueva, de la cual, nos interesa encontrar el color rojo, para ello es necesario generar una aproximación de la intensidad de la componente de la imagen que posee los valores del color de interés.

Al poseer ya la aproximación de la intensidad del componente del color rojo de la imagen, para asegurar que los valores tomados son los correctos, se genera un umbral, para este caso se utilizan los valores mayores a 40, si fueran inferiores a este, podría tomar tonalidades de rojo que no corresponden a los objetos a examinar.

Al poseer una matriz donde solo se encuentran los valores deseados, se dispone a encontrar el centro de tales valores, para esto se utiliza la función `find`, al poseer las coordenadas `xy` del centro del color rojo en la imagen, se dibuja un pequeño cuadro que se utiliza para visualizar donde está situado el centro del objeto que se desea encontrar.

Es muy importante recalcar que para operar la imagen, la clase de los datos que se generen debe estar en `uint8` (8 bits sin signo), y de la igual manera debe ser la información a la salida del bloque.

```
function [r_out,g_out,b_out] = fcn(r_int,g_int,b_int)

imagenrgb=zeros(640,480,3);
imagenrgb(:,:,1)=r_int;
imagenrgb(:,:,3)=g_int;
imagenrgb(:,:,2)=b_int;
Imagen = uint8(imagenrgb);
```

```

Red = r_int - g_int/2 - b_int/2;
bw = Red > 40;

[x, y] = find(bw);
if ~isempty(x) && ~isempty(y)
    xm = round(mean(x));
    ym = round(mean(y));
    xx = max(1, xm-5):min(xm+5, size(bw, 1));
    yy = max(1, ym-5):min(ym+5, size(bw, 2));
    bwbw = zeros(size(bw), 'uint8');
    bwbw(xx, yy) = 255;

    imagenpro=zeros(640,480,3);
    imagenpro(:,:,1)=bwbw;
    imagenpro = uint8(imagenpro);

    Imagen = Imagen + imagenpro;
    Imagen = uint8(Imagen);

end

r_out = Imagen(:,:,1);
g_out = Imagen(:,:,3);
b_out = Imagen(:,:,2);

```

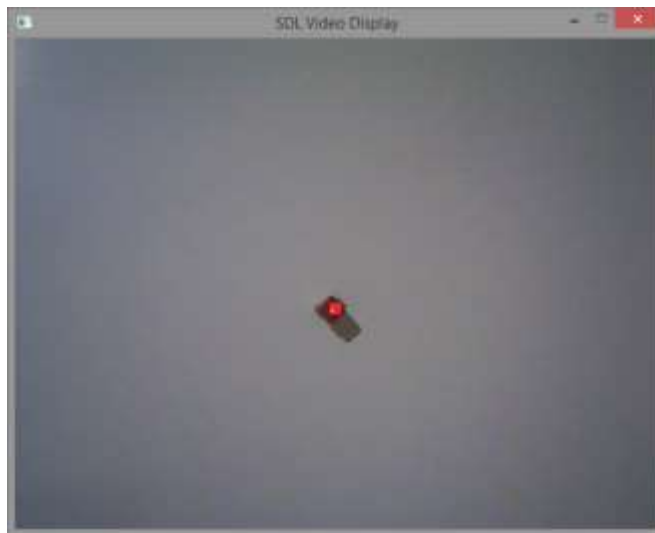


Figura 59. Imagen de salida del bloque de video con reconocimiento de color rojo.

En la misma prueba se modifica el código para que solo se muestre a la salida del bloque, el centro del objeto encontrado.

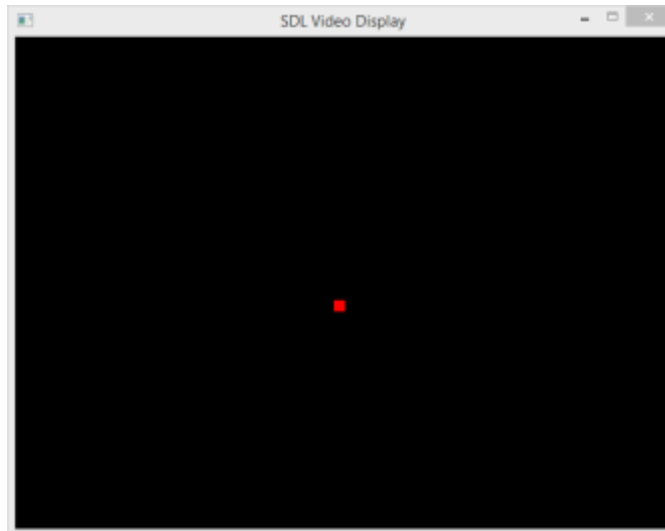


Figura 60. Reconocimiento de color rojo sin imagen de fondo.

Prueba 7. Extraer información a Workspace.

Se utiliza el bloque de 'To Workspace' para lograr extraer información de la simulación, esto con el fin de conocer la posición del objeto a seguir tras el tratamiento de imagen efectuado en el bloque de 'Matlab Function'. Esto se efectúa para poder observar el movimiento del elemento en un determinado tiempo.

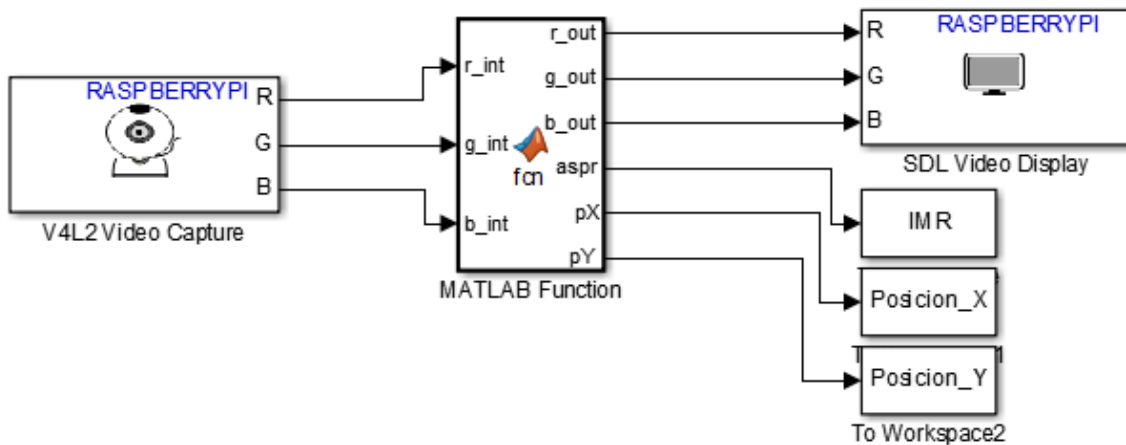


Figura 61. Captura y tratamiento de imagen con el bloque Matlab Function y envié de datos a Workspace.

Name ▲	Value	Min	Max
IMR	640x480x8 uint8	<Too ...	<Too ...
Posicion_X	[0;318;317;317;317;31...	0	318
Posicion_Y	[0;264;264;263;264;26...	0	265

Figura 62. Imagen de Workspace con los datos entregados por Simulink.

Después de trabajar la simulación utilizando los bloques disponibles en simulink para el procesamiento de imágenes y utilizando el bloque de ‘Matlab Funtion’, se logró observar que al usar únicamente bloques, la respuesta de la simulación al ser cargada en la Raspberry pi no era eficiente respecto al tiempo de procesamiento y respuesta, dado que los bloques consumen más recursos al efectuar dichos procesos. Mientras que la simulación utilizando el bloque de ‘Matlab Funtion’ entregó una respuesta más eficiente en términos de tiempo de procesamiento y exactitud en los datos a procesar. La única complicación que genera el uso del código, es la limitada utilización de funciones que ofrece Matlab cuando se ejecuta la simulación en la Raspberry pi.

A partir de las pruebas anteriores para la captura y le procesamiento de imágenes, se puede decir que la opción más acertada para llevar a cabo este proceso es, aquella en la que se usa la menor cantidad de bloques y se utilizan eficientemente líneas de código y funciones que sean compatibles entre las plataformas.

4.2.2 *Procesamiento de Datos*

El procesamiento de datos está dado por dos etapas, la primera será la encargada de calcular la posición del proyectil en los ejes X y Y, la segunda se encarga de utilizar la información referente a la posición del proyectil para encontrar los datos de interés, estos datos serán: la velocidad inicial, la velocidad inicial en cada componente, el ángulo de disparo, posición final del proyectil, altura máxima que alcanza el proyectil y el tiempo de desplazamiento.

a. Coordenadas de desplazamiento

Para hallar las coordenadas de desplazamiento del proyectil, se utilizó la información conseguida en las pruebas realizadas en la captura de imagen, las cuales buscaban identificar cual era el método más eficiente para utilizar Simulink según las necesidades de la experimentación. En la figura 63, se muestra el sistema de bloques a utilizar en la parte de captura de información y entrega de coordenadas de la trayectoria del proyectil. Se tienen dos tipos de bloques para la entrega de información, los 'To Workspace' y 'To File'. Los primeros se encargan de enviar la información al workspace de Matlab, los segundos se encargan de generar un archivo .mat que se encontraran en la raspberry pi. En el código del primer bloque de 'Matlab Function' (Anexo 2), se procesó la información y posteriormente se enviaron los datos del movimiento en los ejes X y Y de la trayectoria del proyectil. Estos datos son enviados uno por uno y los bloques que entregan esa información los guardan como vectores. El tamaño de estos vectores está dado por la cantidad de información que es posible capturar y procesar en un intervalo de 5 segundos.

El código del segundo bloque de 'Matlab Function' se encarga de separar la matriz del color en el cual se encuentra la trayectoria del proyectil, y que es guardada y enviada al workspace. Esta información queda guardada como una matriz de 480 x 640 pixeles, y el largo depende de la información que se capture en el proceso. Al reproducir como video la matriz de información guardada se encontró que solo guarda 20 fps.

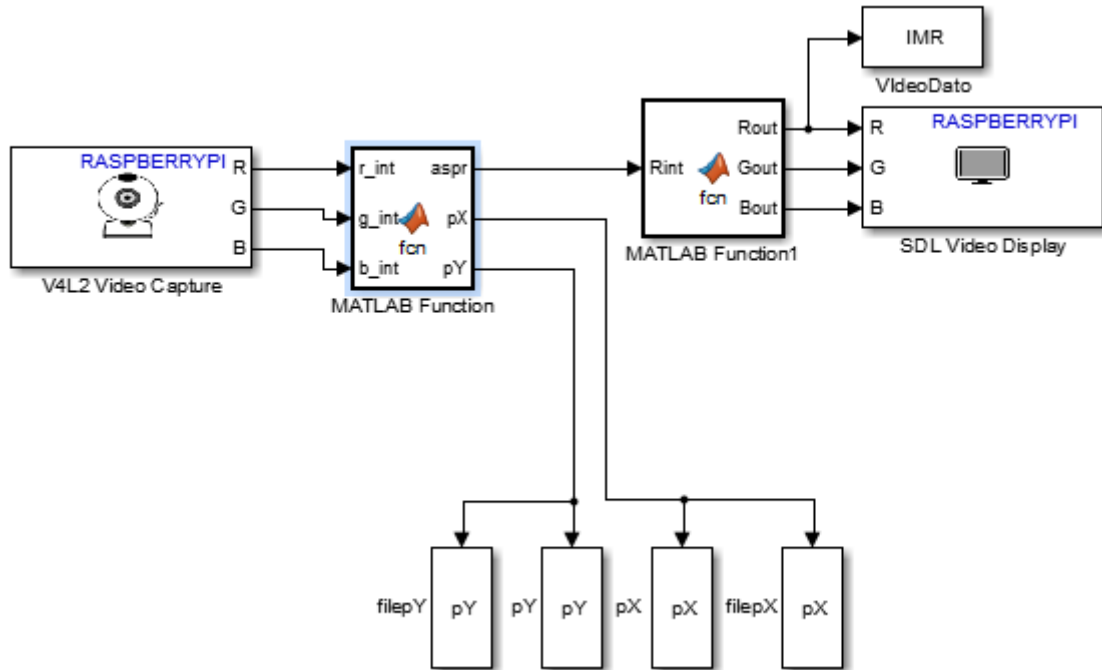


Figura 63. Sistema de bloques en Simulink para captura y procesamiento de imagen.

El sistema de bloques de Simulink y los códigos utilizados para la captura de información son compilados y enviados a la Raspberry por medio de una conexión Lan, esto genera un tiempo de retardo entre el empaquetado, envío de información, inicio del programa y retroalimentación. Se puede observar en la Figura 64, parte del protocolo que utiliza Matlab en el momento de correr el programa.

```

Administrador: C:\Windows\SYSTEM32\cmd.exe
**starting the node1**
** starting the model **
**stopping the model**
**Simulation finished**
C:\Users\andrenfelipe\Desktop\Rasp_mat1\captura_imagen_rtt>

```

Figura 64. Ventana con protocolos utilizados por Matlab y la Raspberry Pi.

b. Procesamiento de información

El procesamiento de información (figura 65), necesita inicialmente de los vectores que guardan los datos de la trayectoria del proyectil, tanto del eje X como el Y. Estos datos son entregados utilizando el bloque 'From File' que busca un archivo .mat, y carga el vector completo en el bloque 'Matlab Function'. Si se utiliza el bloque de 'From workspace', se envían los datos del vector uno a uno, y para el lograr procesar y calcular los datos de interés para la experimentación, no es posible trabajar si no se tienen los vectores completos.

En el Anexo 3, se encuentra el código utilizado para el procesamiento de la información, el cual funciona principalmente utilizando la función Polyfit, la cual hace un ajuste polinómico de segundo grado del vector que posee los datos del movimiento en el eje Y, y entrega como resultado la velocidad inicial del proyectil del componente del eje Y. Para lograr utilizar la función Polyfit, es necesario conocer el tiempo de duración del desplazamiento, para lo cual se tiene el tamaño del vector de datos y sabiendo que el componente de tiempo está sujeto a los 20 fps, logramos despejar y conocer que cada salto de tiempo será de 0.05 segundos, entonces el vector de tiempo está dado entre los 0 segundos y el tamaño del vector por 0.05 segundos. Al poseer los datos de la trayectoria del proyectil, se puede calcular por Pitágoras el ángulo con el cual fue disparado el proyectil, y utilizando la ecuación $V_{yi} = V_i \sin \Theta_i$ se despeja la velocidad inicial del proyectil. Al poseer la velocidad inicial del proyectil, la velocidad inicial de cada componente y la trayectoria del desplazamiento del proyectil, es posible calcular la información faltante y enviarla al workspace para lograr mostrarla al usuario utilizando una GUI (interfaz gráfica de usuario).

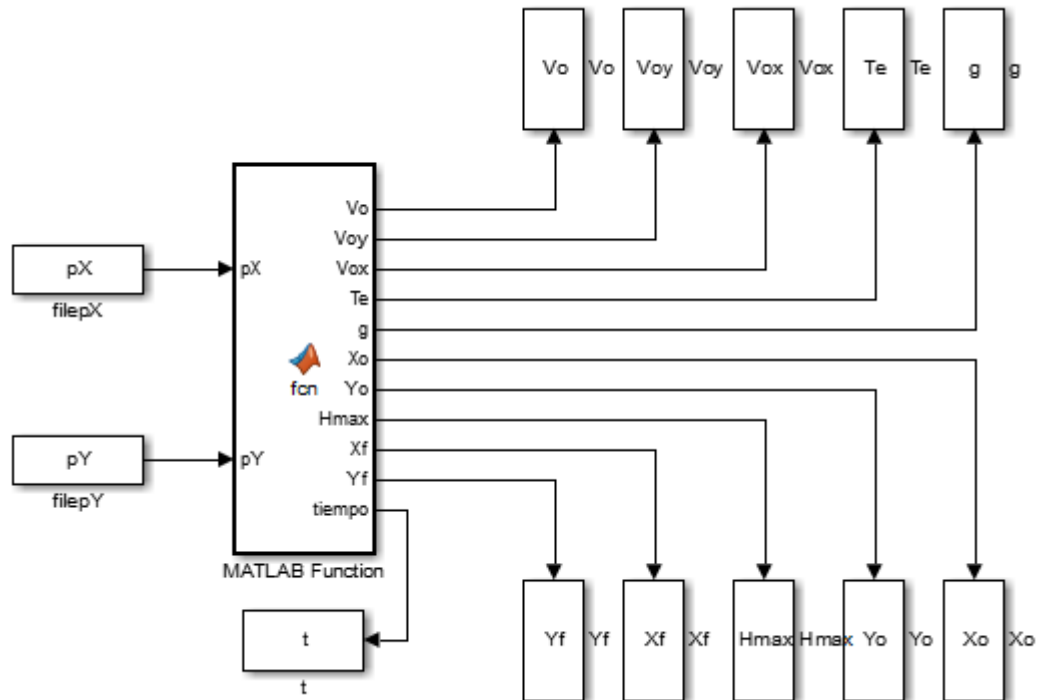


Figura 65. Sistema de bloques en Simulink para el procesamiento y entrega de datos.

c. Entrega de datos

Para lograr que el usuario tenga facilidad a la hora de la experimentación se utilizó una GUI programada desde Matlab. Esta GUI brinda al usuario la posibilidad de manejar las dos etapas de procesamiento de información, tal como se ve en la figura 66. El primer pulsador de la plataforma llamado “iniciar captura”, se encarga de ejecutar la primera etapa, para capturar imagen y procesa las mismas, por medio de la Raspberry pi y enviar la respectiva información al workspace. El segundo pulsador se llama “calcular datos”, y se encarga de cargar los archivos .mat, donde se encuentran los datos del desplazamiento del proyectil para procesarlos y entregar la información de interés del movimiento parabólico. Un tercer pulsador llamado “visualizar trayectoria”, se encarga de reproducir en forma de video la matriz entregada por la primera etapa, para poder comparar el desplazamiento con los datos obtenidos tal como se ve en la figura 69, y por último se cuenta con un pulsador llamado “resetear”, el cual se encarga de limpiar la información de la GUI.

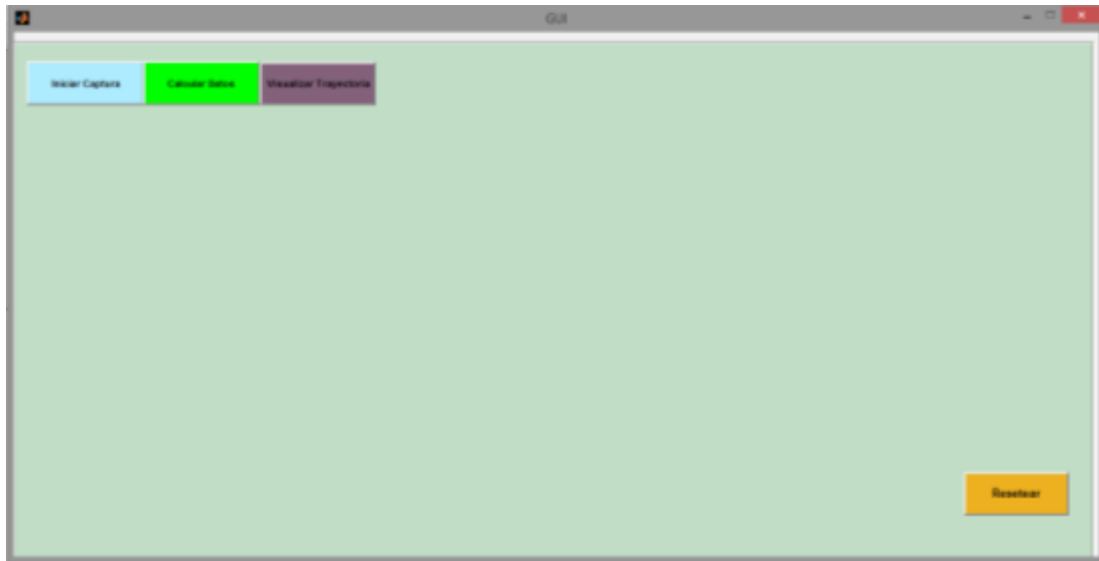


Figura 66. GUI sin ejecutar datos.

Al tener los datos de la trayectoria del proyectil guardados, y ejecutar la etapa de procesamiento de datos, se entregara la información del desplazamiento del objeto en forma de tabla, como se puede observar en la figura 67.



Figura 67. GUI con la tabla de posición del proyectil.

Al finalizar el procesamiento de los datos al utilizar la etapa de cálculo, se muestra en la plataforma los resultados obtenidos y la gráfica respectiva del desplazamiento del proyectil, tal como se muestra en la figura 68.

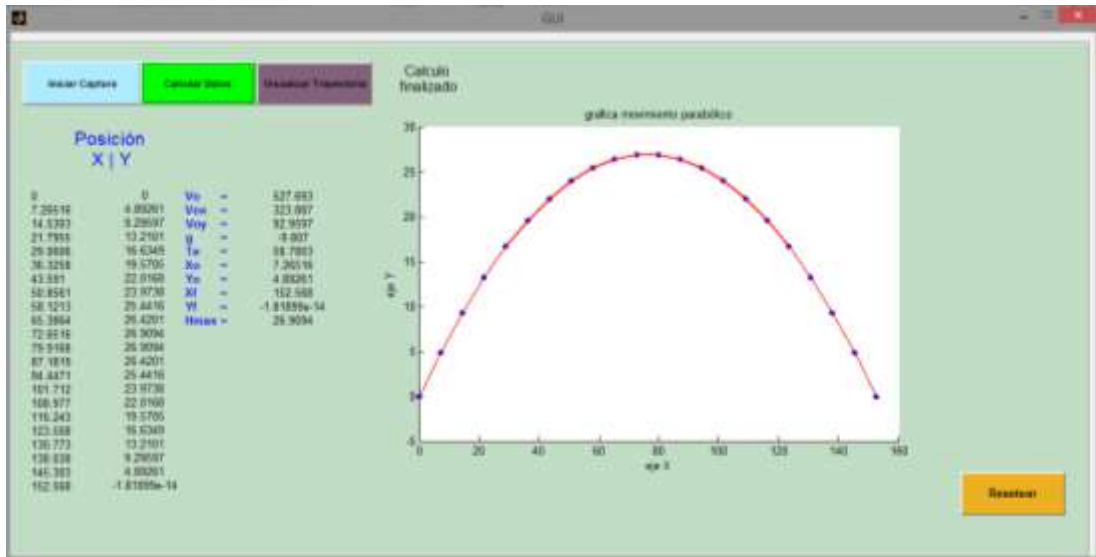


Figura 68. GUI con todos los datos en pantalla.

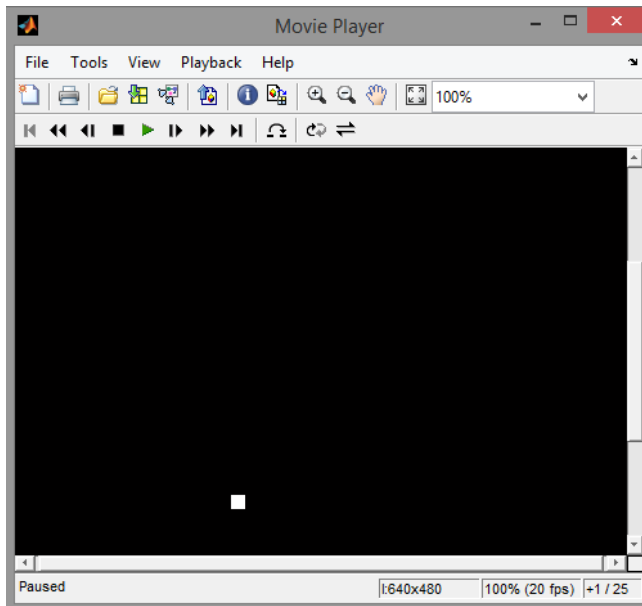


Figura 69. Reproductor de video de Matlab, para mostrar el desplazamiento del proyectil guardado en una matriz.

5 PRUEBAS Y RESULTADOS

Al realizar las pruebas de la estructura en general, se logró evidenciar:

- a. En cuanto al sistema de recarga del proyectil.

Al activar la recarga del proyectil, se identificaron casos particulares de su funcionamiento. El sistema posee un tiempo de activación necesario para que las plataformas de transporte del proyectil logren llevarlo desde una posición inicial hasta el sistema de disparo, y que las plataformas puedan quedar ubicadas en una posición específica en espera del nuevo proyectil.

Para transportar el proyectil desde el sistema de recarga hasta el disparador, éste se apoya en la pared derecha de la estructura hasta llegar a la parte superior. Esta pared debe tener un mínimo de fricción con el fin de no frenar el proyectil durante el proceso y hacer que este caiga antes de llegar al disparador. Al reducir la fricción de la pared utilizando cera, el transporte del proyectil es exitoso, pero el tiempo en que la plataforma recoge, transporta el proyectil y regresa a su posición inicial varía. Como el tiempo necesario para transportar el proyectil es variable dependiendo de la fricción de la plataforma con el proyectil, la plataforma genera un error de posición de 2 cm.

En la parte superior de la estructura existe una canaleta que se encarga de posicionar el proyectil por encima del sistema de disparo para lograr efectuar la experimentación, si el sistema de recarga no tiene la suficiente fuerza a la hora de subir el proyectil, este se quedará frenado en la canaleta y no se efectuara el disparo, por el contrario, si lleva más fuerza de la necesaria, se genera un efecto catapulta haciendo que el proyectil sea lanzado fuera de la estructura, para lo cual se dejó un pwm de 65, que equivale a una alimentación de motor de 3.05 voltios.

b. En cuanto al sistema de ajuste de altura.

El sistema de ajuste de altura al poseer tres acciones diferentes, subir, bajar y dejar inmóvil la plataforma de disparo, posee dos barras de soporte que le ayudan al desplazamiento, además mantienen centrada la plataforma. Sin embargo, el movimiento de la plataforma genera vibración, que puede variar dependiendo de la velocidad con la cual se desplace la plataforma. No obstante, esto no genera ninguna alteración en las pruebas, debido a que el posicionamiento de este sistema se realiza antes de iniciar la experimentación.

La plataforma de disparo se puede desplazar en un intervalo de 0 cm a 38 cm de altura, si la plataforma supera los 38 cm, el sistema de disparo chocará contra la superficie de la estructura, de igual manera si la plataforma intenta ubicarse por debajo de los 0 cm, la plataforma se frenará contra la estructura y se afectará la unión del motor con el tornillo sin fin.

c. En cuanto al sistema de ajuste de ángulo.

El sistema de ajuste de ángulo es manipulado desde un servo motor, que se encuentra unido al sistema de disparo en uno de sus extremos. En la unión del servo motor y el sistema de disparo se generó un espacio que afecta el movimiento de ajuste de ángulo, haciendo que al variar el sentido en el cual se va a disparar el proyectil, el ángulo tenga un error de 10° . Por tal razón, es necesario ubicar el ángulo de disparo 10° sobre el ángulo desde el cual se desea disparar.

d. En cuanto al sistema de disparo.

El sistema de disparo cuenta con siete diferentes niveles de velocidad recomendados para la experimentación, los cuales están ajustados para que el mínimo nivel de velocidad sea el necesario para que el proyectil pueda ser disparado, y el máximo nivel de la posibilidad de que el proyectil llegue hasta el extremo izquierdo de la estructura. Estos

niveles fueron elegidos desde la variación de un pwm, con valores desde el 10 como menor potencia y 70 como máxima, haciendo que las variaciones de voltaje sean de 0.47 voltios hasta los 3.3 voltios. Al aumentar esta alimentación el proyectil es disparado con demasiada fuerza lo que genera un choque contra la estructura que no permite llevar a cabo la experimentación.

e. En cuanto a las variaciones de altura, ángulo y potencia de disparo.

Se realizaron 68 diferentes pruebas de disparo en la estructura de experimentación, haciendo variaciones en la altura de disparo, la potencia con la que el proyectil es lanzado y el ángulo en el cual se encuentra la boquilla del disparador. Las pruebas fueron distribuidas en 42 disparos con un ángulo de cero grados y variando la altura de la plataforma del disparador entre la posición de 0 cm, 5 cm, 10 cm, 20 cm, 30 cm y 38 cm, siendo esta última la máxima altura en la cual el disparador puede estar, con variaciones de la potencia del disparador en los niveles 1, 3 y 7. Con una altura de la plataforma de 0 cm, se realizaron 11 diferentes lanzamientos, los cuales tenían variaciones de 15°, 30°, 45°, 60° y 65°, ajustando la potencia de disparo entre los niveles 3 y 7. Manejando una altura de 10 cm, se ejecutaron 8 diferentes pruebas, donde se variaron los ángulos del disparador entre los -30°, 45° y 65°, con variaciones de potencia de disparo en los niveles 1, 3 y 7. Y a una altura de 30 cm se efectuaron 7 distintos lanzamientos, donde estos tenían variaciones en los ángulos de -30°, 45° y 65°.

Número Prueba	ángulo del disparo	altura del disparador	potencia de disparo
1	0°	0 cm	1
2	0°	0 cm	2
3	0°	0 cm	3
4	0°	0 cm	4
5	0°	0 cm	5
6	0°	0 cm	6
7	0°	0 cm	7
8	0°	5 cm	1
9	0°	5 cm	2

10	0°	5 cm	3
11	0°	5 cm	4
12	0°	5 cm	5
13	0°	5 cm	6
14	0°	5 cm	7
15	0°	10 cm	1
16	0°	10 cm	2
17	0°	10 cm	3
18	0°	10 cm	4
19	0°	10 cm	5
20	0°	10 cm	6
21	0°	10 cm	7
22	0°	20 cm	1
23	0°	20 cm	2
24	0°	20 cm	3
25	0°	20 cm	4
26	0°	20 cm	5
27	0°	20 cm	6
28	0°	20 cm	7
29	0°	30 cm	1
30	0°	30 cm	2
31	0°	30 cm	3
32	0°	30 cm	4
33	0°	30 cm	5
34	0°	30 cm	6
35	0°	30 cm	7
36	0°	38 cm	1
37	0°	38 cm	2
38	0°	38 cm	3
39	0°	38 cm	4
40	0°	38 cm	5
41	0°	38 cm	6
42	0°	38 cm	7
43	15°	0 cm	3
44	15°	0 cm	7
45	30°	0 cm	3
46	30°	0 cm	7
47	45°	0 cm	3
48	45°	0 cm	7
49	60°	0 cm	3
50	60°	0 cm	7

51	65°	0 cm	1
52	65°	0 cm	3
53	65°	0 cm	7
54	-30°	10 cm	1
55	-30°	10 cm	3
56	-30°	10 cm	7
57	45°	10 cm	1
58	45°	10 cm	3
59	45°	10 cm	7
60	65°	10 cm	3
61	65°	10 cm	7
62	-30°	30 cm	1
63	-30°	30 cm	3
64	-30°	30 cm	7
65	45°	30 cm	1
66	45°	30 cm	3
67	45°	30 cm	7
68	65°	30 cm	3

Tabla 5. Pruebas de disparo haciendo variaciones en ángulo del disparador, altura de plataforma de disparo y potencia de disparo.

De las pruebas de disparo realizadas se tomaron las de mayor importancia en la experimentación, al entregar la mayor cantidad de información de relevancia respecto a la estructura y **parámetros** de física.

➤ **Pruebas 1, 2, 3, 4, 5, 6, 7:**

Las anteriores pruebas se realizaron con una altura de 0 cm en la plataforma de disparo y un ángulo de 0°, haciendo variaciones en la velocidad de disparo de nivel 1 al nivel 7.

Al efectuar la prueba 1, se observó un error de posición en la plataforma de disparo, la cual genera un error de 3 mm respecto a la base. Al ser disparado el proyectil este se desplaza de manera recta pero rebota contra la superficie, aunque este efecto puede ser despreciable para la toma de datos, dada su reducida alteración en la posición respecto al eje vertical. Al aumentar el nivel de potencia de disparo, se evidenció un aumento en los rebotes que se tienen del proyectil con respecto a la superficie de la plataforma. También

se logra evidenciar que al aumentar el efecto de rebote del proyectil, la dirección en la cual se desplace el proyectil puede variar, y eso afecta la toma de datos de manera significativa sabiendo que la importancia de la experimentación respecto a la toma de datos, es la posición en la cual se desplace el proyectil.

También se pudo observar que al aumentar el nivel de la potencia de disparo se logra evidenciar un aumento en la distancia recorrida del proyectil en un tiempo determinado, esto demuestra que la velocidad del desplazamiento del proyectil es mayor al aumentar la potencia de disparo, pero al superar el nivel 5, el proyectil choca contra la estructura y se ve afectada la dirección de su desplazamiento, afectando la captura de información.





Figura 70. Pruebas 1 a la 7, aumento en la velocidad de desplazamiento del proyectil.

➤ **Prueba 36, 37, 38, 39, 40, 41, 42:**

Las anteriores pruebas se realizaron con una altura de 38 cm en la plataforma de disparo y un ángulo de 0° , haciendo variaciones en la velocidad de disparo de nivel 1 al nivel 7.

Al efectuar las pruebas 36, 37, 38, 39, 40, 41 y 42, se observó que al ser lanzado el proyectil, este genera una proyección horizontal. El punto en el cual el proyectil impacta en la estructura aumenta respecto al eje horizontal, al aumentar el nivel en la potencia del disparador. A partir del nivel 6 de velocidad del disparador, el proyectil alcanza a chocar contra la pared lateral izquierda de la estructura, esto afecta la toma de datos, dado que el proyectil genera un rebote que puede variar la dirección en la cual se desplace el proyectil dentro de la estructura.

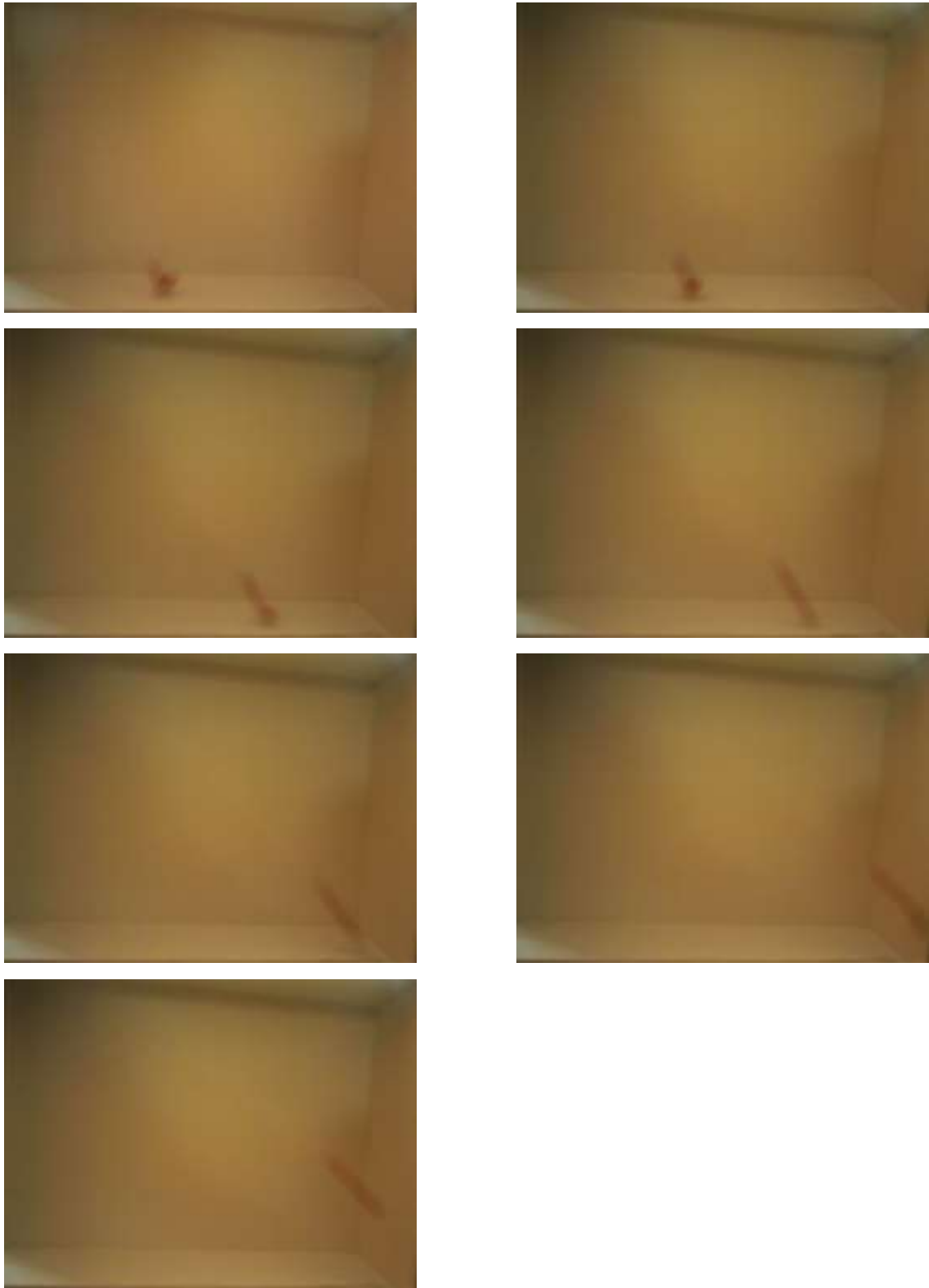


Figura 71. Desplazamiento del proyectil respecto al punto de impacto al aumentar la potencia de disparo en las pruebas 36 a la 42.

➤ **Prueba 47, 48:**

Las anteriores pruebas se realizaron con una altura de 0 cm en la plataforma de disparo y un ángulo de 45° , haciendo variaciones en la velocidad de disparo en el nivel 3 y en el nivel 7.

Al generar el disparo del proyectil, se logra evidenciar un desplazamiento parabólico, el cual muestra una velocidad constante en el desplazamiento sobre el eje horizontal, por tal motivo la aceleración en este sentido es cero. Por otra parte, se observa un cambio en la velocidad respecto al eje vertical, esto demuestra el efecto de la gravedad sobre el proyectil. Al observar los desplazamientos de las dos pruebas, se logra visualizar que el cambio en la potencia de disparo afecta significativamente la distancia recorrida por el proyectil en el eje horizontal, y el cambio de la posición de la altura máxima en el eje vertical a la cual puede llegar el proyectil.



Figura 72. Desplazamiento con un nivel 3 en el disparador.



Figura 73. Desplazamiento con un nivel 7 en el disparador.

➤ **Prueba 68:**

La anterior prueba se realizó con una altura de 30 cm en la plataforma de disparo y un ángulo de 65° , y una velocidad de disparo en el nivel 3.

Al analizar el movimiento efectuado por el proyectil en esta experimentación, se logra evidenciar un desplazamiento parabólico, el cual al alcanzar la altura en la cual fue lanzado, empieza a comportarse con una proyección horizontal. Se logra observar que el proyectil experimenta una aceleración constante hacia abajo debido a la fuerza gravitacional, y esto hace que la velocidad vertical cambie uniformemente.

Es importante recalcar que la experimentación fue realizada solo con la potencia de disparo a nivel 3, dado que a menos potencia no se obtiene mucha información y a más potencia el proyectil chocará contra la parte superior de la estructura, afectando la toma de datos para la experimentación si se necesita observar una trayectoria parabólica.



Figura 74. Desplazamiento del proyectil con un nivel 3 en el disparador a 30 cm de la superficie y a 65° .

➤ **Prueba 45, 49:**

La anterior prueba se realizó con una altura de 0 cm en la plataforma de disparo y con los ángulos de 30° y 60° , y una velocidad de disparo en el nivel 3.

Se comparan estas dos pruebas, dado que el ángulo con el cual son disparados los proyectiles, son ángulos complementarios. Esto en el momento de la experimentación

muestra que el punto en el cual los proyectiles chocan en el eje horizontal, debe ser los mismos, sabiendo que la potencia de disparo es igual para ambas pruebas, la altura del disparador la misma y los ángulos de disparo deben ser complementarios, como es el caso.



Figura 75. Disparo a 30°



Figura 76. Disparo a 60°

Distancia máxima alcanzada en el eje horizontal por los dos proyectiles al ser disparados con ángulos complementarios.

Prueba del GUI

Al desarrollar las pruebas del software para la captura y procesamiento de información, se utilizó la Interfaz Gráfica de Usuario (GUI) diseñada para la facilidad del usuario en poder manipular los diferentes sistemas de procesamiento que se tuvieron en cuenta para esta etapa del trabajo. La GUI posee cuatro pulsadores, el primero activa la captura de video por 5 segundos y envía la información al workspace de Matlab, el segundo pulsador calcula los datos de importancia de la experimentación, el tercero genera una ventana de video para reproducir la información capturada, y por último, el cuarto pulsador reinicia la GUI.

El primer sistema está basado en la captura de información por medio de una cámara web y procesamiento de imágenes con el fin de entregar la posición del eje X y el eje Y. La captura de esta información se da en 5 segundos, dado que es un tiempo considerable para el desplazamiento del proyectil, el inconveniente de este proceso se da cuando la captura se está registrando con intervalos de 0.25 segundos, y la información recopilada

en la matriz se comprime a 20 fps, esto genera que la captura y procesamiento de información sea poco fiable para la experimentación.

El segundo sistema es el de procesamiento y entrega de las condiciones del movimiento parabólico. En esta etapa el software necesita de los datos del primer sistema para funcionar. Al tener los datos de posición en los ejes X y Y del desplazamiento del proyectil, se calcula el tiempo de desplazamiento y se genera una aproximación de la velocidad inicial y sus componentes, también muestra en la pantalla de la GUI la información que tiene y una gráfica según los datos obtenidos del movimiento del proyectil. Si los datos de la trayectoria del objeto entregados por la primera etapa son erróneos, el resultado de la segunda etapa será erróneo también.

De igual manera que el segundo sistema, al ejecutar el tercer pulsador, si no se ha ejecutado el sistema de captura de imagen, no podrá ser mostrado ningún video, y si la toma de datos fue errónea o los datos están mal, el video a mostrar será diferente a lo que se espera ver en la experimentación.

El cuarto pulsador se encarga de reiniciar el sistema, lo que hace es dejar en no visible todos los elementos que se encuentran en la pantalla, esto con el fin de poder efectuar una nueva experimentación.

6 GUÍA DE USUARIO

Este dispositivo para experimentación cuenta con dos elementos, el primero es una estructura que le permite al usuario efectuar el disparo de un proyectil para observar la trayectoria, y lograr efectuar diferentes tipos de experimentación en física cinemática, y más específicamente en movimiento parabólico. El segundo elemento es un software que fue diseñado como una GUI (Interfaz Gráfica de Usuario) para efectuar la captura de

información y procesamiento de imagen, y entrega de información acerca del desplazamiento del proyectil.

La estructura cuenta con un sistema de disparo al cual se le puede variar la altura desde los 0 cm hasta los 38 cm, ajustar el ángulo de posición del disparador desde los -30° hasta los 65° , y modificar la potencia de disparador en siete diferentes posiciones recomendadas para la experimentación. Estas variables son controladas desde una consola de mando, que cuenta con tres perillas y un pulsador, cada perilla controla una de las variables, la primera a mano izquierda controla el sistema de ajuste de ángulo, la perilla del medio controla la altura de la plataforma de disparo, girándola en sentido de las manecillas del reloj subirá, en contra de las manecilla descenderá y en el centro la plataforma estará estática, y por último la perilla a mano derecha controla la potencia con la cual el proyectil será disparado. Cuenta también con un pulsador, el cual controla un sistema de recarga que posee una plataforma que transporta el proyectil hasta el sistema de disparo, luego de efectuar el lanzamiento del proyectil, este retorna a la plataforma de recarga por medio de una canaleta dispuesta a un extremo de la estructura.

El uso apropiado de la estructura desde la consola está dado por orden jerárquico:

1. Ajuste de ángulo.
2. Altura de la plataforma de disparo.
3. Potencia del disparador.
4. Recarga del proyectil.

Esta dispuesto de esta manera para logra un correcto funcionamiento de todos los elementos y un orden en los datos de la experimentación.

El software fue diseñado como una GUI para facilitarle al usuario el manejo de los elementos. Este cuenta con cuatro pulsadores, los cuales sirven para activar el sistema de captura de imagen, procesamiento de información y entrega de datos al usuario.

El primer pulsador se encarga de activar el sistema de captura de imagen y al terminar la captura, entrega al software la información de la posición del objeto en el transcurso de la experimentación. Es importante verificar que los datos de captura sean correctos, para lo cual se tiene un tercer pulsador, el cual abre una ventana de video, donde se puede observar el desplazamiento del proyectil. El segundo pulsador se encarga de tomar los datos de posición del proyectil y calcular los valores de interés para la experimentación, estos valores son: Velocidad inicial del proyectil, los componentes de la velocidad inicial de los ejes X y Y, posición inicial y final de los componentes de la trayectoria, valor de la gravedad y altura máxima del proyectil en la componente Y.

7 PROYECCIONES

Para la entrega del dispositivo didáctico para laboratorio de cinemática con sistema embebido, se planteó la elaboración de un control de disparo que maneje posición, fuerza de disparo y recarga automática de un proyectil, todo esto dentro de un espacio con las condiciones apropiadas para la experimentación, además de esos elementos, también se planeó el diseño de un software que permita la adquisición de datos por medio de sensores, el procesamiento de los mismos y una visualización de los datos procesados.

Se puede observar el trabajo como el compendio de:

- 1) Dispositivo didáctico
 - Sistema para ajustar el ángulo del disparador.
 - Sistema para controlar la altura del disparador.
 - Sistema para modificar la potencia del proyectil.
 - Sistema que permite la recarga del proyectil.
 - Estructura para el montaje de los diferentes sistemas.
 - Estructura con las condiciones para la experimentación.
 - Consola para el control de los diferentes sistemas.
- 2) Software para procesos de imagen
 - Software y elementos para la adquisición de imagen.

- Software para procesamiento de datos.
- Visualización de datos.

De los dos componentes que se nombraron anteriormente, se logró desarrollar e implementar del primer elemento una estructura que cumple con las condiciones apropiadas para desarrollar una experimentación en física cinemática, la cual cumple con un disparador que permite modificar su posición vertical, ajustar el ángulo en el cual se desea disparar el proyectil y variar la potencia con la cual es lanzado, un sistema que permite la recarga del proyectil en el disparador, y una estructura donde están instalados los diferentes componentes adecuadamente y un espacio apropiado en el cual se desplace el proyectil en el momento de la experimentación.

Del segundo componente se logró desarrollar e implementar un software para la adquisición de imagen y reconocimiento de objeto para identificar la posición en la que se encuentra el proyectil a la hora de la experimentación. De la información recopilada por la captura, se hace una aproximación polinómica de la parábola para calcular la velocidad de los componentes, de igual manera se logra calcular la información pertinente para la experimentación y entregar los datos al usuario en una GUI.

Problemas y posibles soluciones

Se logró evidencias después de desarrollar las pruebas pertinentes en la estructura, que al utilizar el sistema de recarga del proyectil, en el momento de soltar el objeto en el disparador, se desarrolla un efecto catapulta con la plataforma, que está sujeta a la cadena plástica de transporte y esto genera un lanzamiento no deseado del proyectil, que puede afectar en el proceso de disparo y experimentación. Puede ser solucionado el problema modificando la estructura, al ubicar el eje superior de soporte de la cadena de transporte del sistema de recarga del proyectil dos centímetros más arriba.

En el momento de que el proyectil vaya a ser disparado por el dispositivo, este puede caer por el borde de la estructura que recarga el disparador, en este momento si la plataforma no está nivelada, el proyectil no entrara al disparador y esto afectara la experimentación. Es necesario revisar la estructura y utilizar algún tipo de material que

logre eliminar cualquier fricción de la plataforma con las varillas de soporte, y agregar un nuevo anclaje a la plataforma con el tornillo sin fin.

Al variar el ángulo de disparo sobre el eje horizontal y colocar el proyectil en el disparador, si el proyectil no tiene la suficiente fuerza, este no será disparado y quedará embotellado en el dispositivo. Para solucionar este problema se propone colocar en la parte superior de la estructura un elemento que aumente el impulso con el que cae el proyectil en el disparador, para que logre ser arrojado.

El proyectil al ser disparado, puede chocar contra las paredes internas y quedar frenado en algún punto de la estructura. Se recomienda dar un mayor grado de inclinación en la parte interna inferior de la cámara de observación para lograr una correcta evacuación del proyectil.

El sistema de captura de información entrega los datos correctamente, el mayor inconveniente está en la cámara web que se utilizó, la cual debería proporcionar una velocidad de captura de imágenes de 90 fps, pero al efectuar la experimentación con el software se evidencio que la captura no era mayor a 15 o 20 fps. Para lo cual se recomienda cambiar la cámara web por una de mayor velocidad de captura y confiabilidad. También se puede cambiar el sistema de disparo por uno que utilice un proyectil más grande y de igual manera, el espacio diseñado para la experimentación sea más amplio, esto con el fin de que el sistema de captura de información logre identificar el proyectil y calcular su posición en el espacio de dos ejes.

Aportes personales

La estructura para experimentación fue diseñada pensando en el movimiento de proyectiles, sea de tipo parabólico o media parábola. Las aplicaciones en la experimentación no se limitan solo en este tipo de experimentación, este dispositivo brinda la posibilidad de poder experimentar otro tipo de movimientos o laboratorios que pueden ser de utilidad o ayuda en un aula de clase, entre los cuales se encuentran:

- Movimiento en una dimensión (eje horizontal).
- Choque elástico.
- Caída libre con sucesivos rebotes.

Además, la estructura y el software fueron pensados para la experimentación con desplazamiento parabólico, esto da la posibilidad de ajustar el sistema para las nuevas experimentaciones.

8 CONCLUSIONES

En primer lugar, se puede concluir que se logró e implementar una estructura que cumple con las condiciones apropiadas para desarrollar la experimentación en física cinemática. A su vez, se logró desarrollar e implementar un software para la adquisición de imagen y reconocimiento del objeto, del cual se calcula la información pertinente para la experimentación y entrega al usuario los datos de tal experimentación en una GUI.

Luego de realizar diferentes pruebas a la estructura y al software se puede concluir, además, que:

Este dispositivo constituye un material de apoyo, el cual facilita la presentación de la información, la manipulación de las variables, la observación del proceso y el análisis de los resultados.

Es importante hacer pruebas para calibrar los diferentes sistemas mecánicos que son accionados por motores, dado que si no tienen la suficiente fuerza o tiene más de la requerida para realizar la acción, puede generar un mal funcionamiento de todo el sistema.

Al realizar las experimentaciones se evidenció que la realización de prácticas a pequeña escala exigen equipos de mayor precisión, por lo tanto las pruebas realizadas en el proceso

de captura y procesamiento de información son poco confiables. Para mejorar las prácticas de laboratorio es pertinente mejorar los equipos de captura de información o hacer la experimentación a mayor escala.

Al trabajar con una metodología de prototipado, es necesario tener los elementos para realizar pruebas y cambios en los sistemas que se estén desarrollando, dado que siempre aparecerán inconvenientes a la hora de la construcción o empalme con otros sistemas.

A partir de las experimentaciones realizadas con este trabajo se identificaron otros posibles usos de este dispositivo en la práctica como lo son: el movimiento en una dimensión (eje horizontal), choque elástico y caída libre con sucesivos rebotes.

9 BIBLIOGRAFÍA

Arduino Uno. Arduino. Consultado marzo de 2015. Disponible en:
<http://www.arduino.cc/en/Main/ArduinoBoardUno>

C. Platero, “Apuntes de Visión Artificial”. Consultado Octubre de 2015. Disponible en:
http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf

G. Molinero, “Segmentación de imágenes en color basada en el crecimiento de regiones” Biblioteca de la Universidad Distrital Francisco José de Caldas. Consultado Octubre de 2015. Disponible en:
<http://revistas.udistrital.edu.co/ojs/index.php/visele/article/view/3568/5549>

G. González; C. Eduardo (2008). Los métodos de enseñanza en el proceso de enseñanza aprendizaje de las Ciencias Exactas. V Congreso Internacional Didáctica de las Ciencias, Ciudad de la Habana.

J. Jiménez; B. Canacue. (2012). Kit didáctico de laboratorio para la enseñanza de movimiento unidimensional y planos inclinados con fricción. Universidad Pedagógica Nacional, Bogotá, Colombia.

J.I. Pozo. (1989). Teorías cognitivas del aprendizaje. Morata. España, Madrid.

J.I. Pozo. (1996). Aprendices y maestros. La nueva cultura del aprendizaje. Alianza. España, Madrid.

Morales, A. Briceño, E. & Mena, J. (2011). El rol de las gráficas en la construcción de modelos. Pontificia Universidad Católica de Valparaíso, Chile

Piezas de Fischertechnik. Consultado Octubre de 2015. Disponible en: <http://www.fischertechnik.de/en/PortalData/1/Resources/produkte/091082/091082-einzelteilliste.pdf>

Ramos, C. (2010). Seguimiento visual de objetos utilizando técnicas de predicción. Universidad politécnica de Valencia

Ramírez, Y. (2010). Prácticas de laboratorio de ciencias naturales, para el colegio de bachilleres del estado de Querétaro, de las asignaturas de física i y física ii plantel Ezequiel Montes no.11. Escuela Superior de Ingeniería Mecánica y Eléctrica, México. D.F

Raspberry Pi B. Raspberrry. Consultado septiembre de 2014. Disponible en: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

R. E. Woods; R. C. González; S. L. Eddins. (2009). Digital image processing using Matlab. Gonzalez, Woods, & Eddins en: F. Giraldo; M. Gonzales; E. Camargo. (2011). Algoritmos de procesamiento de imagen satelitales con transformada Hough. Revista Visión Electrónica.

R. C. González; P. Wintz. (1996). Procesamiento digital de imágenes. Addison-Wesley en: F. Giraldo; M. Gonzales; E. Camargo. (2011). Algoritmos de procesamiento de imagen satelitales con transformada Hough. Revista Visión Electrónica.

Raymond, A. Serway & John, W. Jewett, Jr. (2008). Física para ciencias e ingenierías 7° edición. Cengage Learning Editores, México, D.F

Sobrado, M. Eddie, A. (2003). Sistemas de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Pontificia universidad católica del Perú escuela de graduados, Perú

T. Acharya; A. K. Ray. (2005). Image processing: principles and applications. John Wiley & Sons en: F. Giraldo; M. Gonzales; E. Camargo. (2011). Algoritmos de procesamiento de imagen satelitales con transformada Hough. Revista Visión Electrónica.

Tipler, M. (2001). Física para la ciencia y la tecnología 5° Edición. Editorial reverté. España, Barcelona.

Universidad de Jaén, “Transformada De Hough (HGT)”. Consultado Octubre de 2015. Disponible en http://www4.ujaen.es/~satorres/practicas/practica4_vc.pdf

H. Youg; G. Freedman. (2009). Física Universitaria 10° Edición. Person Educación, México, D.F

Anexo 1. Código Arduino

```
#include <HerkuleX.h>

#define RX 8 // Servo Motor TX Pin
#define TX 9 // Servo Motor RX Pin
#define MOTORID 253 // ID Servo Motor

// Salidas digitales de los motores
int MotorD = 5;
int MotorR = 6;
int MotorA1 = 10;
int MotorA2 = 11;

// Entradas análogas para manejo de motores
int pulsador = 0;
int Potenciometro_1 = 3;
int Potenciometro_2 = 4;
int Potenciometro_3 = 5;

// Variables internas
int datoR = 0;
int datoD = 0;
int Valor1 = 0;
int Valor2 = 0;
int Valor3 = 0;
int angulo = 0;

void setup() {
  HerkuleX.begin(115200, RX, TX);
  delay(10);
  // Torque ON
  HerkuleX.torqueOn(MOTORID);
}

void loop() {
```

```

////////////////////////////////////
// Angulo
Valor1 = analogRead(Potenciometro_1);
dato1 = map(Valor1, 0, 1023, -30, 65);
HerkuleX.movePos(MOTORID, 512, angulo, HERKULEX_LED_BLUE);
////////////////////////////////////
// Altura
Valor2 = analogRead(Potenciometro_2);
if(Valor2 < 341){
  analogWrite (MotorA1,0);
  analogWrite (MotorA2,140); // sube
}
if(Valor2 > 341 && Valor2 < 682){
  analogWrite (MotorA1,0);
  analogWrite (MotorA2,0);
}
if(Valor2 > 682){
  analogWrite (MotorA1,130); // baja
  analogWrite (MotorA2,0);
}
////////////////////////////////////
// Disparador
Valor3 = analogRead(Potenciometro_3);
datoD = map(Valor3, 1023, 0, 10, 70);
analogWrite (MotorD,datoD);
////////////////////////////////////
// Recarga de proyectil
datoR = analogRead(pulsador);
if(datoR > 1020){
  analogWrite (MotorR,65);
  delay(5650);
}

```

```
analogWrite (MotorR,0);  
datoR = 0;  
}  
}
```

Anexo 2. Código 'Matlab Function' captura de imagen. Entrega de datos X/Y.

```
function [aspr,pX,pY] = fcn(r_int,g_int,b_int)
```

```
imagenrgb=zeros(640,480,3);  
imagenrgb(:,:,1)=r_int;  
imagenrgb(:,:,3)=g_int;  
imagenrgb(:,:,2)=b_int;  
Imagen = uint8(imagenrgb);
```

```
Datos_x = 0;  
Datos_y = 0;
```

```
Red = r_int - g_int/2 - b_int/2;  
bw = Red > 40;
```

```
[x, y] = find(bw);  
if ~isempty(x) && ~isempty(y)  
    xm = round(mean(x));  
    ym = round(mean(y));  
    xx = max(1, xm-5):min(xm+5, size(bw, 1));  
    yy = max(1, ym-5):min(ym+5, size(bw, 2));  
    bwbw = zeros(size(bw), 'uint8');  
    bwbw(xx, yy) = 255;
```

```
imagenpro=zeros(640,480,3);  
imagenpro(:,:,1)=bwbw;  
imagenpro = uint8(imagenpro);
```

```
Imagen = imagenpro;  
Imagen = uint8(Imagen);
```

```
Datos_x = xm;  
Datos_y = ym;
```

```
end
```

```
aspr = Imagen(:,:,1);  
pX = Datos_x/10.666666;  
pY = Datos_y/10.666666;
```

Anexo 3. Código 'Matlab Function' Procesamiento de información. Entrega de datos.

```
function [Vo,Voy,Vox,Te,g,Xo,Yo,Hmax,Xf,Yf,tiempo]= fcn(pX,pY)
```

```
X = pX';
```

```
Y = pY';
```

```
D = length(Y);
```

```
tiempo = D*0.05;
```

```
t = 0:0.05:tiempo-0.05;
```

```
pol_y= polyfit(t,Y,2);
```

```
Voy = pol_y(2);
```

```
g =-9.807;
```

```
Hip0 = sqrt((X(2)-X(1))^2+(Y(2)-Y(1))^2);
```

```
Ady0 = X(2)-X(1);
```

```
Te = (asin(Ady0/Hip0))*180/pi;
```

```
Hmax = 0;
```

```
for T=1:D-1
```

```
    if Y(T) > Hmax
```

```
        Hmax = Y(T);
```

```
    end
```

```
end
```

```
Vo = (Voy - g*(33.0012))/sin(Te);
```

```
Vox = Vo*cos(Te);
```

```
Xo = X(1);
```

```
Yo = Y(1);
```

```
Xf = X(D);
```

```
Yf = Y(D);
```

Anexo 4. Código GUI en Matlab.

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%   GUI, by itself, creates a new GUI or raises the existing
%   singleton*.
%
%   H = GUI returns the handle to a new GUI or the handle to
%   the existing singleton*.
%
%   GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI.M with the given input arguments.
%
%   GUI('Property','Value',...) creates a new GUI or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to GUI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 05-Nov-2015 11:36:31

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @GUI_OpeningFcn, ...
                  'gui_OutputFcn', @GUI_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```

% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

% Choose default command line output for GUI
handles.output = hObject;

set(handles.text1,'String','')
set(handles.text2,'String','')
set(handles.text3,'String','')
set(handles.text4,'Visible','off')
set(handles.text5,'String','')
set(handles.text6,'Visible','off')
set(handles.axes1,'Visible','off')

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%-----botones
Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text1,'String','Espere Por Favor')
set_param('captura_imagen','SimulationCommand','start')
set(handles.text1,'String','Iniciando Captura')
pause(7)

```

```
set(handles.text1,'String','Captura finalizada')
```

```
% --- Executes on button press in pushbutton2.
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton2 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
set(handles.text4,'String',' Posición X | Y')
```

```
set(handles.text4,'Visible','on')
```

```
global X Y Datos
```

```
X = evalin('base','pX');
```

```
set(handles.text2,'String',X)
```

```
Y = evalin('base','pY');
```

```
set(handles.text3,'String',Y)
```

```
save('pX.mat','X');
```

```
save('pY.mat','Y');
```

```
set(handles.text1,'String','Iniciando Calculo')
```

```
set_param('calculo','SimulationCommand','start')
```

```
pause(5)
```

```
set(handles.text1,'String','Calculo finalizado')
```

```
Vo = evalin('base','Vo');
```

```
Voy = evalin('base','Voy');
```

```
Vox =evalin('base','Vox');
```

```
Vox = -Vox;
```

```
g = evalin('base','g');
```

```
Te = evalin('base','Te');
```

```
Xo = evalin('base','Xo');
```

```
Yo = evalin('base','Yo');
```

```
Hmax = evalin('base','Hmax');
```

```
Xf = evalin('base','Xf');
```

```
Yf = evalin('base','Yf');
```

```
Datos = [Vo Vox Voy g Te Xo Yo Xf Yf Hmax];
```

```
set(handles.text5,'String',Datos)
```

```
set(handles.text6,'Visible','on')
```

```
t = evalin('base','t');
```

```
T = 0:0.05:t;
```

```
set(handles.axes1,'Visible','on')
```

```
axes(handles.axes1)
```

```
hold on
```

```
plot(X,Y,'xb','linewidth',5)
```

```
plot(X,Y,'r','linewidth',2)
```

```
title('grafica movimiento parabólico')
```

```
xlabel('eje X');
```

```
ylabel('eje Y');
```

```
grid
```



```

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global M
M = evalin('base','IMR');
implay(M)
set(handles.text1,'String','trayectoria del proyectil')

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
set(handles.text1,'String','')
set(handles.text2,'String','')
set(handles.text3,'String','')
set(handles.text4,'Visible','off')
set(handles.text5,'String','')
set(handles.text6,'Visible','off')
set(handles.axes1,'Visible','off')
cla

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject handle to axes1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

%-----cajas de texto
% --- Executes during object creation, after setting all properties.
function text1_CreateFcn(hObject, eventdata, handles)
% hObject handle to text2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function text2_CreateFcn(hObject, eventdata, handles)
% hObject handle to text2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(hObject, eventdata, handles)

```

```
% hObject handle to text3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function text4_CreateFcn(hObject, eventdata, handles)
% hObject handle to text4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function text5_CreateFcn(hObject, eventdata, handles)
% hObject handle to text5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function text6_CreateFcn(hObject, eventdata, handles)
% hObject handle to text6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```