

GRADO EN INGENIERIA EN TECNOLOGIA INDUSTRIAL
TRABAJO FIN DE GRADO

**CONSTRUCCION Y CONTROL DE UNA
MAQUETA DE UNA GRUA PORTUARIA**

Alumno: Muñoz Aseguinolaza, Asier

Director: Gómez Garay, Vicente

Curso: 2018-2019

Fecha: Bilbao, 19.07.2019

Resumen trilingüe

Resumen

Título: Construcción y control de una maqueta de una grúa portuaria.

En este documento se expone el diseño de la maqueta de la grúa portuaria Carola, la construcción de esta y el control, utilizando el kit Lego Mindstorms NXT. Este proyecto ha sido realizado como Trabajo de Fin de Grado, en el Departamento de Automática y Control. A lo largo de este documento se describen las diferentes etapas, el diseño y construcción de la maqueta, la programación del sistema de control y el diseño del controlador PID. Además, se ha añadido el presupuesto del proyecto y las conclusiones.

Laburpena

Izenburua: Portuko garabi baten maketaren diseinua eta kontrola.

Dokumentu honetan Carola garabiaren maketaren diseinua aurkezten da, baita honen eraikuntza eta kontrola ere, Lego Mindstorm NXT kit-az baliatuz. Proiektu hau Gradu Amaierako Lan-a izan da, Automatika eta Kontrol sailean egin. Dokumentu honetan zehar proiektuaren etapa desberdinak deskribatzen dira, hala nola maketaren diseinua eta kontrola, edota kontrol sistemaren programazioa eta PID kontrolagailuaren diseinua. Gainera, proiektuaren aurrekontua erantsi da, baita honen ondorioak ere.

Abstract

Title: Construction and control of a model of a port crane.

The design of the model of Carola port crane is presented in this document, as well as its construction and control, using Lego Mindstorms NXT kit. This project has been carried out as End of Degree Work, done on the Department of Automation and Control. Throughout this document its different stages are described, as well as the design and control of the model, or the programming of the control system and the design of the PID controller. Besides, the project's budget has been added, also its conclusions.

Índice

1.-Memoria	7
1.1.-Introducción.....	7
1.2.-Objetivo y alcance del trabajo	8
1.3.-Contexto	9
1.4.-Beneficios del trabajo.....	13
1.5.-Análisis detallado de la grúa Carola	14
1.6.-Análisis de alternativas y elección	16
1.6.1.-Análisis del material disponible	16
1.6.2.-Alternativas posibles y elección	16
1.7.-Solución propuesta. Descripción	20
1.7.1.-Maqueta grúa portuaria	20
1.7.2.-Lenguaje de programación	21
2.-Desarrollo del trabajo	22
2.1.-Descripción de las fases del proyecto	22
2.2.-Diagrama de Gantt	23
2.3.-Diseño de la maqueta	26
2.3.1.-Base	26
2.3.2.-Pórtico	27
2.3.3.-Castillete o plataforma giratoria	28
2.3.4.-Cadrlátero articulado o sistema basculante	28
2.3.5.-Sistema de recogida del tirante	30
2.4.-Construcción de la maqueta.....	30
2.5.-Puesta en marcha del entorno de programación.....	33
2.6.-Diseño del sistema de regulación de velocidad.....	37
2.6.1.-Obtención del modelo de la planta.....	37
2.6.2.-Sintonización del controlador PID.....	39
3.-Aspectos económicos.....	49

3.1.-Mano de obra	49
3.2.-Gastos de montaje	49
3.3.-Gastos de Lego Mindstorms	49
3.4.-Gastos de Software	50
3.5.-Presupuesto ejecutado	50
4.-Conclusiones	51
5.-Bibliografía	52
Anexo I: Manual de usuario	53
Anexo II: Manual de programación	54
Anexo III: Manual de instalación	57

Índice de ilustraciones

Ilustración 1: Grúa Carola	9
Ilustración 2: Kit Lego Mindstorms	10
Ilustración 3: Ladrillo NXT	11
Ilustración 4: Sensores NXT	12
Ilustración 5: Servomotor NXT	12
Ilustración 6: Grúa Carola (El Correo)	15
Ilustración 7: Diagrama de Gantt.....	25
Ilustración 8: Diseño base	26
Ilustración 9: Diseño pórtico	27
Ilustración 10: Diseño plataforma giratoria	28
Ilustración 11: Diseño cuadrilátero articulado	29
Ilustración 12: Construcción pórtico.....	30
Ilustración 13: Construcción plataforma giratoria	31
Ilustración 14: Construcción cadrilátero articulado	32
Ilustración 15: Diagrama de flujo control movimiento.....	34
Ilustración 16: Diagrama de flujo control movimiento.....	35
Ilustración 17: Gráfico N_t sin controlador	37
Ilustración 18: Simulación sistema sin controlador	39
Ilustración 19: Sistema de control realimentado	40
Ilustración 20: Controlador PI.....	40
Ilustración 21: Sintonización controlador	41
Ilustración 22: Señal de control	41
Ilustración 23: Respuesta del controlador simulado y real con controlador PI	42
Ilustración 24: Lugar de raices planta	43
Ilustración 25: Lugar de las raices planta+controlador.....	44
Ilustración 26: Controlador PI.....	45
Ilustración 27: Señal de control	45
Ilustración 28: Señal salida simulada	46
Ilustración 29: Señal de salida experimental.....	46
Ilustración 30: Señal control "Autotune"	47
Ilustración 31: Señal control analítico.....	47
Ilustración 32: Salida "Autotune"	48
Ilustración 33: Salida analítico	48

Índice de tablas

Tabla 1: Análisis material	17
Tabla 2: Análisis lenguaje de programación	18
Tabla 3: Mano de obra	49
Tabla 4: Gastos de montaje	49
Tabla 5: Gastos de Lego Mindstorms	49
Tabla 6: Gastos de Software	50
Tabla 7: Presupuesto ejecutado	50

1.-Memoria

1.1.-Introducción

Mediante este documento se presentan y describen todas las etapas necesarias para la Construcción y Control de una maqueta de una grúa portuaria situada junto a la ría de Bilbao al lado del Palacio Euskalduna.

Se comienza desarrollando la memoria del proyecto, donde se contextualiza la relevancia del trabajo en el diseño de la maqueta de la grúa portuaria. Se describen los objetivos principales del trabajo junto con el alcance de este. Se analizan las diferentes alternativas a tener en cuenta a la hora de diseñar, construir y hacer el programa que controle la maqueta. Para concluir la memoria, se realiza una descripción detallada de la solución propuesta.

Para continuar, se presenta la metodología seguida a lo largo del desarrollo del trabajo, donde se especifican las tareas realizadas en el mismo, así como su orden y duración mediante un diagrama de Gantt. Se prosigue con una descripción de los primeros pasos llevados a cabo para el diseño de la maqueta de la grúa, donde se muestra el diseño inicial de los diferentes elementos de los que está constituida nuestra maqueta. También se muestran los pasos seguidos a la hora de la construcción, así como los diferentes contratiempos que han surgido y cuál ha sido la manera de solucionarlos. Tras ello, se realiza el diseño y la programación del sistema de control, encargado del movimiento de la maqueta.

Por último, se analizan los aspectos económicos del proyecto, donde se describe el presupuesto con el que se deberá contar para llevarlo a cabo. También se incluyen las conclusiones obtenidas tras la realización del trabajo y una bibliografía con los documentos que sirven de base para el desarrollo de la solución.

1.2.-Objetivo y alcance del trabajo

El objetivo de este trabajo de fin de grado, de aquí en adelante TFG, es intentar replicar, de la manera más fiel posible, el funcionamiento de la grúa Carola. Para ello se utilizará, en la mayoría de los casos, el kit LegoMindstorms NXT, así como otros materiales adicionales en caso de ser necesarios.

Teniendo en cuenta la dificultad de replicar el funcionamiento exacto de la grúa, la maqueta ejecutará los siguientes movimientos:

- Giro de la plataforma alrededor del eje vertical.
- Movimiento del cuadrilátero articulado o sistema basculante.
- Movimiento vertical del cable que sirve para recoger objetos.

Se van a hacer las siguientes simplificaciones:

- Dado que un ladrillo solo puede llevar acoplados 3 motores y 4 sensores, se suprimirá el movimiento de translación da la grúa.
- Debido al peso de las piezas que se van a levantar en comparación con el de la maqueta, la estructura principal de la grúa irá anclada al suelo.
- Debido al gran tamaño de los motores en comparación con el tamaño de la maqueta, el motor encargado de recoger y soltar el cable que soporta la carga no formará parte de la estructura principal.

1.3.-Contexto

El TFG se lleva a cabo en el departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingeniera de Bilbao. Este trabajo es el paso final para obtener el grado en Ingeniería en Tecnología Industrial.

Para explicar el contexto se empezará por explicar, por una parte, la grúa Carola, y por otra parte el kit de Lego Mindstorms del que se hará uso para la construcción de la maqueta.

La grúa Carola está ubicada en las gradas del antiguo astillero Euskalduna, instalaciones que forman parte del museo marítimo de la ría de Bilbao y donde se exponen, de forma libre, barcos y elementos navales. La grúa Carola es el reclamo simbólico del museo.

La Carola es una grúa cigüeña diseñada por Arturo Llorente Bodegas, y que fue construida entre los años 1954 y 1957 por la empresa Talleres de Erandio convirtiéndose en la máquina elevadora más potente de toda España. Tenía capacidad para levantar 60 toneladas de peso y era utilizada para la construcción de grandes barcos. Funcionó hasta el año 1984 en que se cerraron los astilleros. La grúa quedó en desuso y el ayuntamiento bilbaíno la adquirió y donó, junto al resto de las instalaciones de los astilleros a la Diputación Foral de Bizkaia para que formara parte del museo.

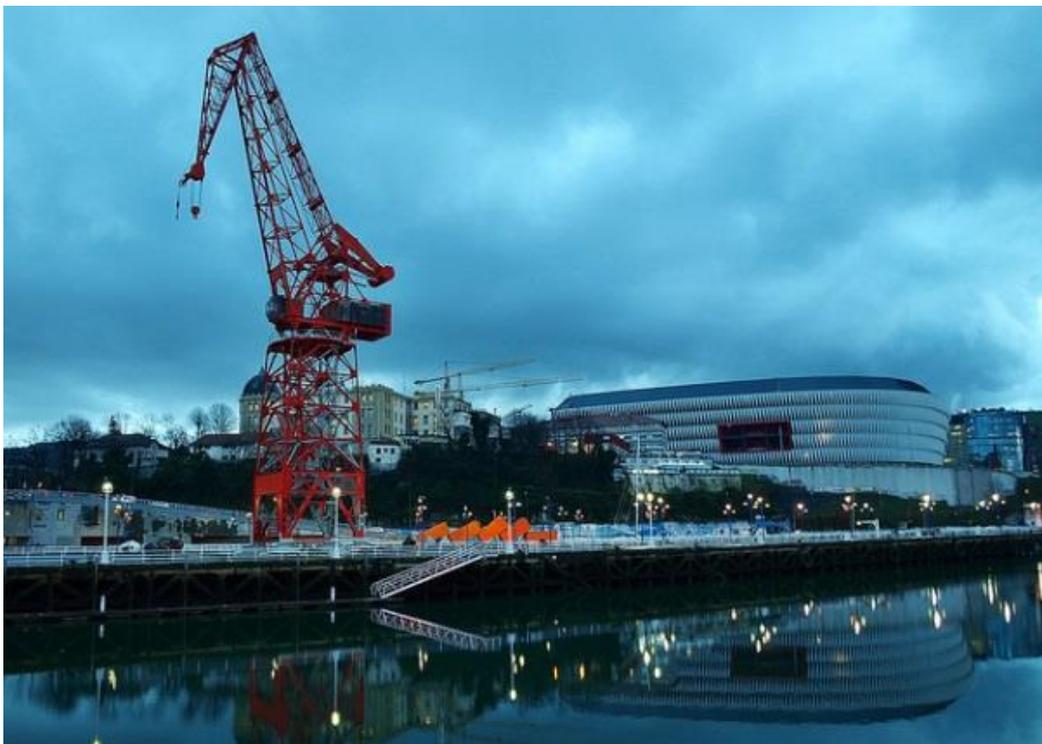


ILUSTRACIÓN 1: GRÚA CAROLA

La construcción se hizo a base de perfiles roblonados, uniendo las chapas que forman la estructura mediante tornillos calentados a 900 °C que eran incrustados con un mazo en su correspondiente lugar. La cabina de mandos está situada a 35 m sobre el suelo y el cuadrilátero a 60 m. Se movía sobre vías para poder desplazarse por el muelle. La Carola debe su nombre a una mujer, Carol Iglesias, que cruzaba la ría desde Deusto para ir a trabajar en las oficinas de hacienda. Tal era el atractivo de la chica que llegaba a parar la producción del astillero cada vez que pasaba. (Wikipedia, la enciclopedia libre, s.f.) [7]

Por otro lado, Lego Mindstorms es una línea de robótica para niños fabricada por la empresa LEGO, que posee elementos básicos de las teorías robóticas, como la unión de piezas y la programación de acciones en forma interactiva. Este robot fue comercializado por primera vez en septiembre de 1998.

Comercialmente se publicita como *Robotic Invention System*, en español Sistema de Invención Robotizado (RIS). También se vende como herramienta educativa, lo que originalmente se pensó en una colaboración entre LEGO y el MIT. La versión educativa se llama *Lego Mindstorms for Schools*, en español Lego Mindstorms para la escuela y viene con un software de programación basado en la GUI de Robolab.



ILUSTRACIÓN 2: KIT LEGO MINDSTORMS

Lego Mindstorms puede ser usado para construir un modelo de sistema integrado con partes electromecánicas controladas por computador. Prácticamente todo puede ser representado con piezas tal como en la vida real, como un elevador o robots industriales.

La simplicidad de uso y la versatilidad del kit LEGO Mindstorms al no requerir conocimientos eléctricos ni electrónicos, ha propiciado que, poco a poco aumente su uso en el campo de la educación.

LEGO Mindstorms es en la actualidad el material de construcción más eficaz para comenzar a experimentar con robots y concentrarse en los aspectos del aprendizaje.

Hasta 2018 ha habido tres generaciones de Lego Mindstorms: el bloque RCX, el bloque NXT y el EV3. (Wikipedia, enciclopedia libre, s.f.) [6]

El material disponible para el desarrollo de este TFG es el kit Lego Mindstorms NXT. Dicho kit está compuesto por ladrillos inteligentes, motores, sensores y diferentes piezas para la construcción de sus diferentes mecanismos. Pueden conectarse diferentes ladrillos entre ellos, así como como con ordenadores mediante bluetooth, cable USB o ethernet.

El hardware de Lego Mindstorms lo componen tres elementos principales:

- **Ladrillo:** El ladrillo del NXT es el cerebro del robot. Es un ladrillo microcontrolado que puede programarse. El ladrillo dispone de un puerto USB, que sirve para hacer la conexión con el ordenador; 3 puertos de salida, para usarlos con servomotores; 4 puertos de entrada para el uso de los sensores; conexión al adaptador de corriente, para cargar la batería del ladrillo; y los botones NXT, usados para navegar a través del menú del ladrillo.



ILUSTRACIÓN 3: LADRILLO NXT

- **Sensores:** Los sensores son un componente indispensable para cualquier maquina que se denomine un robot. Los sensores entregan información sobre el robot y el ambiente con el cual está interactuando. El programa decide que hacer basándose en esa información y en sus propias instrucciones. En el kit Mindstorms NXT se encuentran 4 tipos de sensores: sensor de contacto (Touch), sensor de ultrasonido, sensor de luz y sensor de sonido.

SENSORES

- Sensor ultrasónico
- Sensor de luz
- Sensor de sonido
- Sensor de tacto



ILUSTRACIÓN 4: SENSORES NXT

- **Servomotores:** Proporcionan al robot la capacidad de moverse. Disponen de un sensor de rotación integrado, lo que permite que los movimientos sean muy precisos. Puede medir rotaciones del motor en grados o rotaciones completas, tiene una desviación de $\pm 1^\circ$.

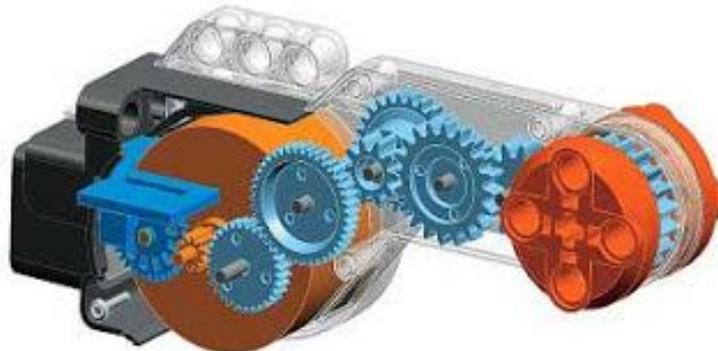


ILUSTRACIÓN 5: SERVOMOTOR NXT

1.4.-Beneficios del trabajo

Los beneficios que aporta el TFG son de diferentes tipos.

Por un lado, tenemos los beneficios sociales. En este caso, el proyecto realizado podría usarse en un futuro para ser expuesto en las jornadas de puertas abiertas de la universidad. Sería una buena manera de hacer ver a un posible futuro alumnado el tipo de cosas con las que podrían llegar a trabajar en el caso de que cursaran este grado. Además, el proyecto también podría ser usado más adelante para que futuros alumnos realizaran su TFG partiendo de este.

Por otro lado, podemos diferenciar los aspectos teóricos o técnicos, en los cuales se observan las diferentes habilidades obtenidas en el transcurso del trabajo. Para la realización de este proyecto ha sido necesario familiarizarse con la construcción de maquetas con el kit de Lego Mindstorms, tanto en lo que se refiere a estructuras como con a los mecanismos. Además, se ha aprendido a realizar un programa para el control del sistema. Finalmente, se han obtenido conocimientos de nuevos métodos de cálculo para la obtención de modelos y estrategias de diseño de controladores mediante herramientas de simulación.

Respecto a los beneficios personales, ha servido como aprendizaje a la hora de realizar un proyecto desde su inicio hasta el final, afrontando los diversos problemas que han ido surgiendo a lo largo de su transcurso y encontrando diferentes soluciones para todos y cada uno de ellos. También se ha aprendido a realizar una planificación del trabajo, a estudiar diferentes materias de forma autónoma, en concreto, el uso de una plataforma y un nuevo lenguaje de programación.

1.5.-Análisis detallado de la grúa Carola

El objetivo de este análisis es estudiar la estructura y el funcionamiento de la grúa de la forma más fiel posible.

Los datos técnicos más importantes de la grúa son los siguientes:

- La grúa tiene una altura máxima de 59 m y la cabina se encuentra a 35 m de altura respecto del suelo.
- La base tiene unas dimensiones de 11 m de largo por 8 m de ancho.
- Tiene una capacidad de elevación de 10 a 20 tm.

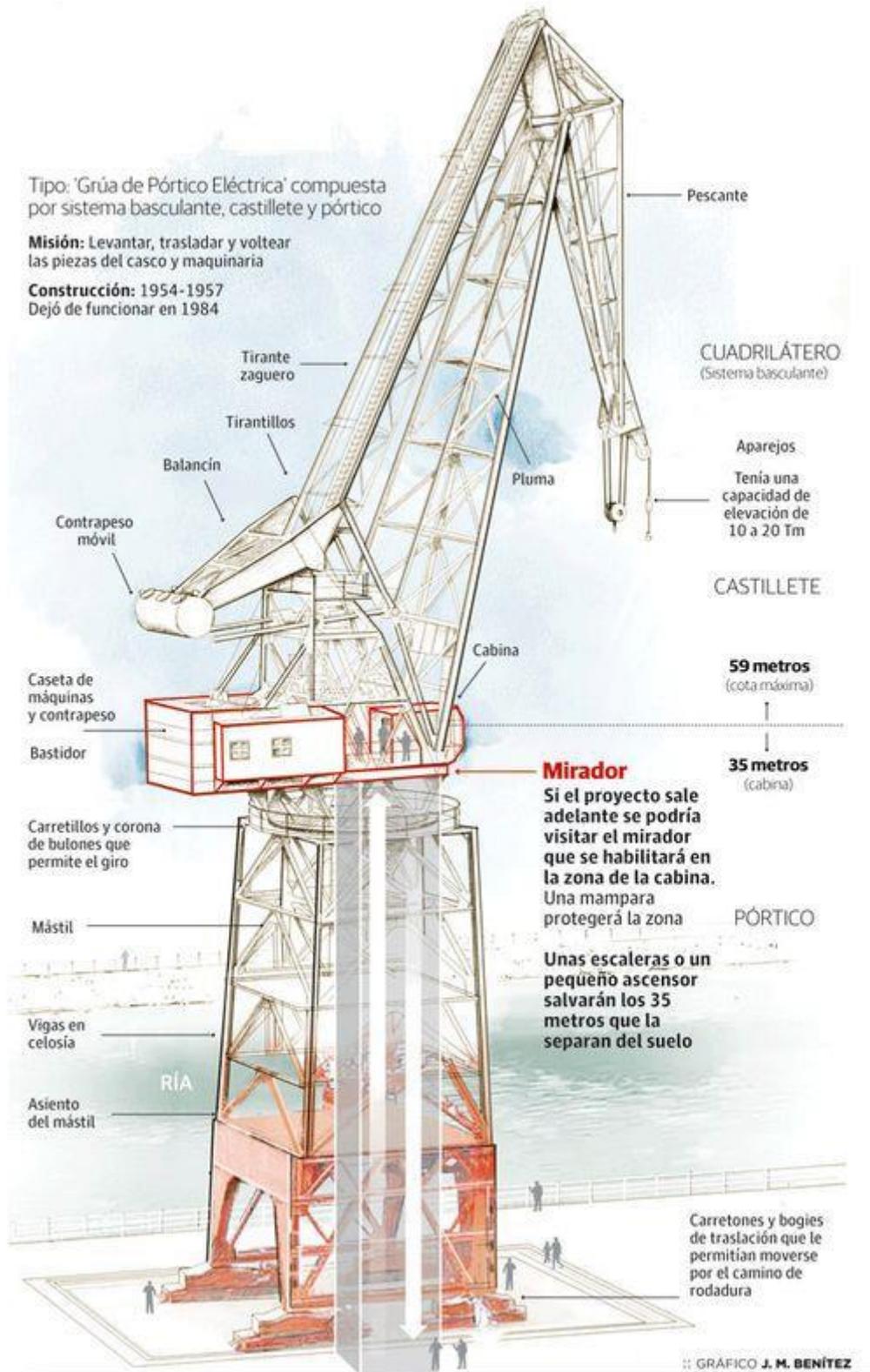


ILUSTRACIÓN 6: GRÚA CAROLA (EL CORREO)

1.6.-Análisis de alternativas y elección

1.6.1.-Análisis del material disponible

En este punto se analiza el material disponible para la construcción de la maqueta.

Por un lado, el kit de Lego Mindstorms, que incluye motores, sensores (contacto, luz, sonido y ultrasonido), que ya se han descrito en un apartado anterior. Además de estos elementos, el kit también incluye elementos de construcción, como vigas, ejes, engranajes etc., además de las piezas clásicas de Lego.

Por otro lado, se pueden utilizar materiales adicionales como madera, metal, tornillos, tuercas, cordeles, etc.

1.6.2.-Alternativas posibles y elección

Este apartado tiene dos partes. Por un lado, la elección del material con el cual se va a construir la maqueta. Por otro lado, la elección del lenguaje de programación que se vaya a usar.

Para valorar y elegir cual es la mejor opción se usa la matriz de priorización, en la cual se ilustran los diferentes aspectos de relevancia, así como el grado de importancia que se dará a cada uno de ellos.

1.6.3.1.-Elección del material

- 1) **Utilizar solamente piezas de Lego.** En esta opción solo se usan las piezas disponibles en el kit Lego Mindstorms como, vigas, engranajes, ladrillos, elementos de unión etc. De esta manera se facilita la unión entre los diferentes elementos de la maqueta. Aun así, debido a que el tamaño de las piezas está preestablecido habrá formas que sean difíciles, incluso imposibles, de construir.
- 2) **Base de metal y resto de Lego.** En esta opción se propone hacer la base de la grúa de piezas de metal y el resto con las piezas de Lego. Tiene la ventaja de que las piezas de metal se pueden cortar a la medida y de esta manera se acoplarán mejor entre ellas. No obstante, presenta el inconveniente de como unir las piezas de metal con el resto de la maqueta que estará compuesta por piezas del kit Lego Mindstorms.

- 3) **Principalmente piezas Lego con algún material adicional.** Esta opción consiste en construir la estructura principal de la maqueta con piezas de Lego, tanto el pórtico principal, como el cuadrilátero articulado. Además, se añadirían diferentes elementos, como una base de madera para fijar la grúa y un electroimán para transportar la carga.

En esta tabla se muestran, por un lado, los criterios de evaluación que han sido seleccionados; por otro lado, la ponderación que recibe cada uno de estos criterios y la puntuación que la asignamos en cada una de nuestras alternativas, que tendrá valores entre 1 y 10.

Los criterios que se van a tener en cuenta son los siguientes: el coste monetario que supone usar dicha alternativa, la posibilidad de elegir las dimensiones de la maqueta, la apariencia estética, la facilidad de montaje y la funcionalidad de los diferentes elementos.

CRITERIOS	PONDERACIÓN	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
COSTES	0,1	9	3	6
DIMENSIONES	0,3	2	8	5
ESTÉTICA	0,1	5	4	7
MONTAJE	0,3	5	4	6
FUNCIONALIDAD	0,2	7	4	6
TOTAL	1	4,9	5,1	5,8

TABLA 1: ANÁLISIS MATERIAL

Teniendo en cuenta los resultados obtenidos, nuestra elección va a ser la tercera alternativa: construcción principalmente con piezas de Lego con algún material adicional.

1.6.3.2.-Elección del lenguaje de programación

- 1) **NXT_G.** Es un método de programación gráfico. No es necesario ningún conocimiento especializado de programación para crear programas relativamente complejos, pero resulta muy incómodo de programar.. (Programmable robots, robotics kit, electronics, education robots - Génération Robots. NXT-G: the development environment supplied with Lego Mindstorms, NXT-G., s.f.) [4]

- 2) **ROBOT-C.** Es lenguaje de programación potente basado en C, en un entorno Windows para escribir y depurar programas. ROBOT-C es una multiplataforma que permite a los estudiantes aprender a programar en un programa basado en C, usado en educación avanzada y aplicaciones profesionales. (programming., Home of the best robot programming language for Educational Robotics. Made for NXT programming and VEX, s.f.) [5]
- 3) **BricxCC.** Es un entorno de desarrollo integrado (IDE) para programar robots de todas las generaciones de Lego Mindstorms. El lenguaje de programación es Not eXactly C (NXC), muy parecido al C. (BricxCommandCenter, s.f.) [2]

Los criterios a tener en cuenta son los siguientes: la facilidad de uso del programa, la memoria necesaria y la rapidez para la ejecución de los programas.

CRITERIOS	PONDERACIÓN	ALTERNATIVA 1	ALTERNATIVA 2	ALTERNATIVA 3
FACILIDAD	0,5	9	3	8
MEMORIA	0,15	6	8	8
RAPIDEZ	0,35	3	9	9
TOTAL	1	6,45	5,85	8,35

TABLA 2: ANÁLISIS LENGUAJE DE PROGRAMACIÓN

Teniendo en cuenta los resultados obtenidos en la tabla anterior la elección va a ser la tercera alternativa: lenguaje de programación NXC.

1.6.3.3.-Elección de la escala

Otro aspecto a tener en cuenta a la hora de la construcción de la maqueta es la escala que se va a usar. Para ello hay diferentes aspectos a tener en cuenta.

En primer lugar, el tamaño de la grúa y su peso tienen una relación directa, es decir, cuanto más grande sea la grúa mayor será su peso. Teniendo en cuenta que los motores de los que disponemos son de tamaño único y que su potencia es limitada, cuanto más grande sea la maqueta mayor peso tendrá y, por consiguiente, mayor oposición pondrá la estructura a los motores.

En segundo lugar, hay que tener en cuenta que cuanto más pequeña sea la maqueta, menos posibilidades de unión tendremos entre las diferentes piezas. Por ello, nos encontraremos con más problemas a la hora de realizar una construcción que refleje fielmente el original.

En último lugar, la estética de la maqueta también es un aspecto importante: hacer la maqueta muy pequeña en comparación al tamaño de los motores empeorará la apariencia estética.

Por los motivos explicados anteriormente, la escala en la que se va a realizar la maqueta es **1:100**, aproximadamente. La escala no será exacta porque las piezas del kit de Lego tienen unas medidas predeterminadas y es muy difícil hacerla del tamaño exacto para que se cumpla la escala propuesta.

1.7.-Solución propuesta. Descripción

En el siguiente apartado se va a describir la solución propuesta. Tanto en el aspecto de construcción de la maqueta, como en el tema del lenguaje de programación.

1.7.1.-Maqueta grúa portuaria

La maqueta está compuesta por los siguientes elementos:

- **Base.** Base de madera sobre la que irá fijada el pórtico de la maqueta mediante tornillos y tuercas. De esta manera, se dotará de estabilidad a la estructura principal.
- **Pórtico.** Zona baja-media de la grúa. Está formada por el pórtico inferior que presenta una figura rectangular formada por los carretes de traslación y el pórtico superior, donde está incrustada la columna que queda rematada por la corona de giro. Está construido completamente con piezas de Lego. Además, en dos de sus lados van situados dos sensores de luz, responsables de mandar la señal al programa que hace que el movimiento sobre el eje vertical de la grúa se detenga.
- **Castillete o plataforma giratoria.** Zona en la que se sitúa la cabina de mando, la caseta de máquinas y el contrapeso fijo. Además, el elemento fijo del cuadrilátero articulado forma parte de él. Completamente construido con piezas de Lego. Sobre esta plataforma giratoria va situado el motor responsable del movimiento del cuadrilátero articulado. Asimismo, en la parte delantera tiene un saliente, cuyo objetivo es disminuir la intensidad de la luz en la parte cercana del sensor cuando la grúa esté llegando al final de su movimiento.
- **Cuadrilátero o sistema basculante.** Formado por el pescante, la pluma, el tirante y el contrapeso móvil con su balancín de contrapeso. La mayoría de sus elementos están contruidos con piezas de Lego. Aunque el tirante lo conforma un cordel fino y su extremo va atado a un electroimán, que servirá para poder coger piezas metálicas de diferentes tamaños.

- **Sistemas de control.** Se utiliza un ladrillo NXT, al que van conectados, mediante cables, los motores y los sensores.

El ladrillo va situado sobre la base de la estructura, en la parte trasera de la grúa, ya que por comodidad y estética, todos los cables salen hacia esa dirección.

Se utilizan tres motores. El primero, situado sobre el pórtico, hará posible el giro del resto de la estructura sobre un eje vertical. El segundo, situado sobre la plataforma giratoria, será el encargado de realizar el movimiento del sistema basculante o cuadrilátero articulado. El tercero y último, encargado de recoger y soltar el cable, va situado en la base de la maqueta. La idea original era situarlo sobre la plataforma giratoria, pero debido al gran tamaño de los motores en comparación con el de la estructura se ha optado por esta solución.

1.7.2.-Lenguaje de programación

El lenguaje de programación que vamos a usar es el NXC. NXC es un lenguaje de alto nivel similar a C. No utiliza el firmware original de Lego, por lo que es necesario reescribir el firmware del ladrillo NXT, que está disponible en la propia plataforma BricxCC.

NXC ha sido desarrollado por John Hansen. Referenciando sus comentarios, el entorno de programación gráfico NXT-G es adecuado si te inicias en el mundo de la programación, mientras que si eres un programador ésta puede ser una alternativa más adecuada.

Nos proporciona una guía del programador, además del compilador y editor, BricxCC. Disponemos además de varios ejemplos.

Como documentación adicional, podemos mencionar la guía en inglés y el libro “Creating Cool Mindstorms NXT Robots”, escritos ambos por Daniele Benedettelli.

Existen además otros proyectos que han sido programados también en NXC y que nos pueden servir como referencia y ayuda. Por ejemplo, los dos diseños de Hans Andersson, tanto el resolovedor del cubo de Rubik, como el resolovedor de sudokus, han sido programados en NXC.

John Hansen ha publicado un libro sobre programación con NBC y NXC: “NXT Power Programming Robotics in C”.

(NXC, s.f.) [3]

2.-Desarrollo del trabajo

2.1.-Descripción de las fases del proyecto

En el desarrollo de este trabajo, se diferencian cuatro fases principales:

- 1) Estudio de viabilidad del proyecto.
- 2) Elección de materiales y diseño.
- 3) Construcción de la maqueta.
- 4) Diseño y programación del sistema de control mediante BricxCC.

La primera tarea es familiarizarse con el uso de kit de Lego Mindstorms para la construcción de estructuras y mecanismos. Para ello se ha hecho uso del libro (“The LEGO MINDSTORMS EV3 Idea Book: 181 Simple Machines and Clever Contraptions”). [1]

Otra de las competencias que se han tenido que adquirir es la capacidad de utilizar el programa BricxCC, con el cual se realizará el control del movimiento de la maqueta.

Además, se precisan conocimientos de Matlab y Simulink, con los cuales se realizará el cálculo de los parámetros del controlador PID que se va a implementar.

Antes de comenzar la fase de diseño, se realizó un estudio de viabilidad que fue la base para la elección de la escala de la maqueta, el material con el que se decidió construirla, y el lenguaje de programación que se iba a usar para realizar su control.

Una vez realizado el estudio de viabilidad, se lleva a cabo el primer diseño, el cual tiene como objetivo principal que la maqueta se parezca lo máximo posible a la grúa real.

Posteriormente, se inicia la construcción de la maqueta, surgiendo diversos problemas a lo largo de la construcción que obligarán a modificar este primer diseño.

A continuación, se realiza la puesta en marcha inicial del control, cuyo objetivo es comprobar que el entorno de programación se conecta adecuadamente a la maqueta, permitiendo obtener todas las funcionalidades, como escritura de programas, descarga y ejecución.

Finalmente se diseña el sistema de control realimentado cuyo objetivo es regular la velocidad de giro de la plataforma y el movimiento de la carga.

2.2.-Diagrama de Gantt

A continuación, se presenta la planificación del proyecto. La fecha de inicio del proyecto es el 18 de febrero de 2019 y la fecha de finalización el 18 de julio de 2019.

A continuación, se describen las tareas realizadas a lo largo del proyecto.

1) **Búsqueda de información**

Recopilación de información sobre el funcionamiento real de la grúa portuaria y sobre el kit Lego Mindsorms.

Duración: 10 días.

2) **Diseño**

Diseño de la maqueta y elección de los materiales para su construcción.

Duración: 6 días.

3) **Construcción de la maqueta**

Diferentes fases de la construcción de la maqueta.

Duración total: 13 días.

a. **Construcción de la base**

Duración: 3 días.

b. **Construcción del pórtico**

Duración: 4 días.

c. **Construcción de la plataforma giratoria**

Duración: 6 días.

d. **Construcción del cuadrilátero articulado**

Duración: 3 días.

e. **Construcción del sistema de recogida del tirante**

Duración: 3 días.

4) **Puesta en marcha del entorno de programación**

Realizar es el control del movimiento de la grúa con la ayuda del programa BricxCC y usando el lenguaje NXC.

Duración: 20 días.

5) **Diseño del sistema de regulación de velocidad de la plataforma**

Diseño de un sistema de control realimentado para la regulación de velocidad de la plataforma.

a. **Obtención del modelo de la planta**

Duración: 7 días.

b. **Sintonización del controlador PID**

Duración: 6 días.

6) **Implementación del electroimán**

Colocar el electroimán y conectarlo mediante un cable a la fuente de alimentación.

Duración: 3 días.

7) **Pruebas experimentales y solución de imprevistos**

Probar el funcionamiento de la maqueta y solucionar cualquier problema que pueda surgir durante su funcionamiento.

Duración: 12 días.

8) **Redacción del documento**

Duración: 26 días.

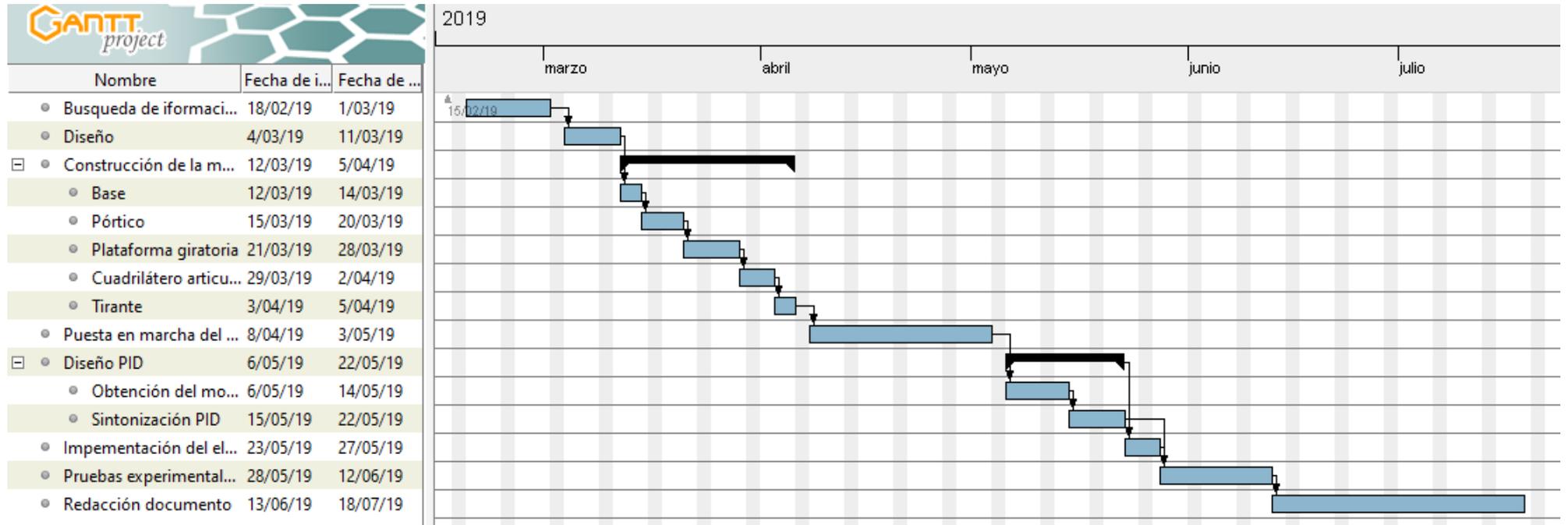


ILUSTRACIÓN 7: DIAGRAMA DE GANTT

2.3.-Diseño de la maqueta

Una vez se ha realizado del estudio de viabilidad y se han analizado todas las alternativas posibles, es el momento de realizar el diseño de la estructura de la maqueta. Para ello se realizarán planos de todos y cada uno de los elementos que componen la maqueta.

2.3.1.-Base

A continuación, se muestra el diseño de la base de la maqueta, en el que se pueden ver cuatro agujeros mediante los cuales vamos a fijar el pórtico a la base. Está fijación se hará mediante unos tornillos y unas tuercas entre la base las vigas de la maqueta.

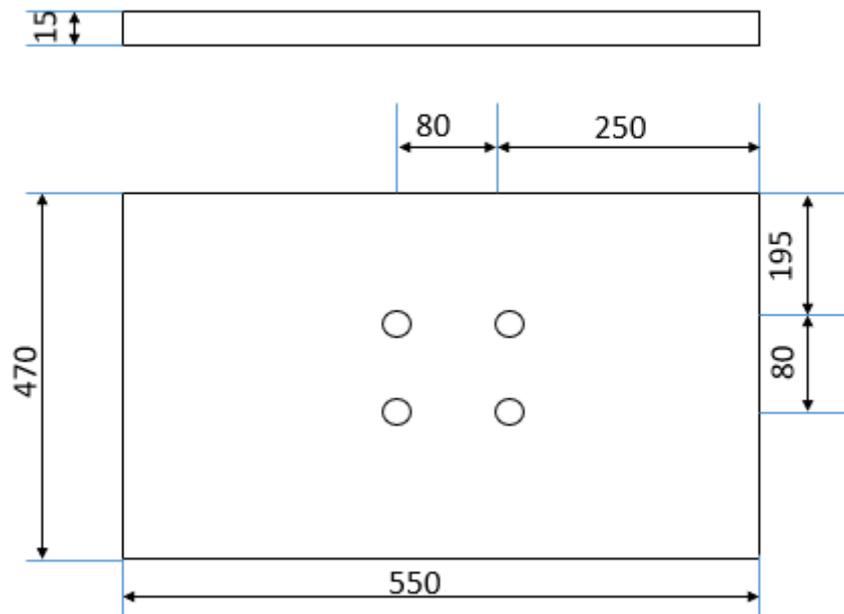


ILUSTRACIÓN 8: DISEÑO BASE

2.3.2.-Pórtico

En la siguiente imagen se puede ver el diseño del pórtico, cuyas barras verticales tienen 15 agujeros, las horizontales 7 y las diagonales 5.

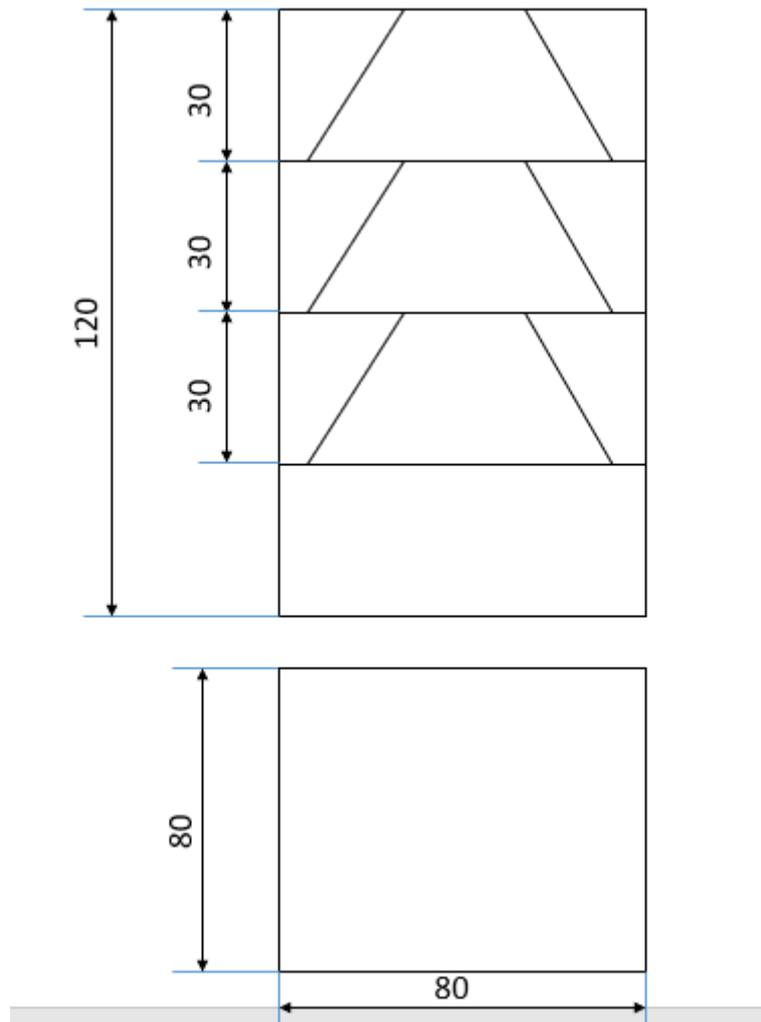


ILUSTRACIÓN 9: DISEÑO PÓRTICO

2.3.3.-Castillete o plataforma giratoria

En un principio se pensó en hacer un diseño simple en el cual la plataforma giraría en torno al eje de un motor que iría fijado al pórtico.

Debido a los problemas de estabilidad experimentados a lo largo de la construcción se decide cambiar este diseño y apostar por uno más estable, aunque también más complejo.

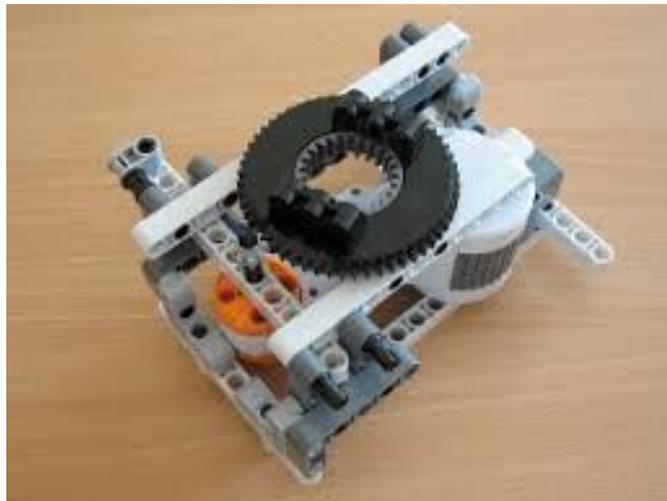


ILUSTRACIÓN 10: DISEÑO PLATAFORMA GIRATORIA

2.3.4.-Cadrilátero articulado o sistema basculante

Para el diseño del cuadrilátero articulado se usa el programa GIM. La razón principal por la que se hace el diseño en este programa es porque queremos que el extremo del pescante, es decir, la punta del cuadrilátero articulado tenga una trayectoria horizontal. Para ello, las distancias se han ajustado para que coincidan con las medidas de las piezas y el mecanismo siga cumpliendo su cometido.

En este caso, las dos barras más largas tienen 15 agujeros pasantes, la tercera tendrá 11 e irá unida a las dos anteriores, y las dos restantes tendrán 7 y 4. La diferencia de altura entre el punto más bajo y el más alto del cuadrilátero es de 150 mm.

A continuación, en la figura 12 con el diseño del mecanismo y la trayectoria que sigue a lo largo de su movimiento.

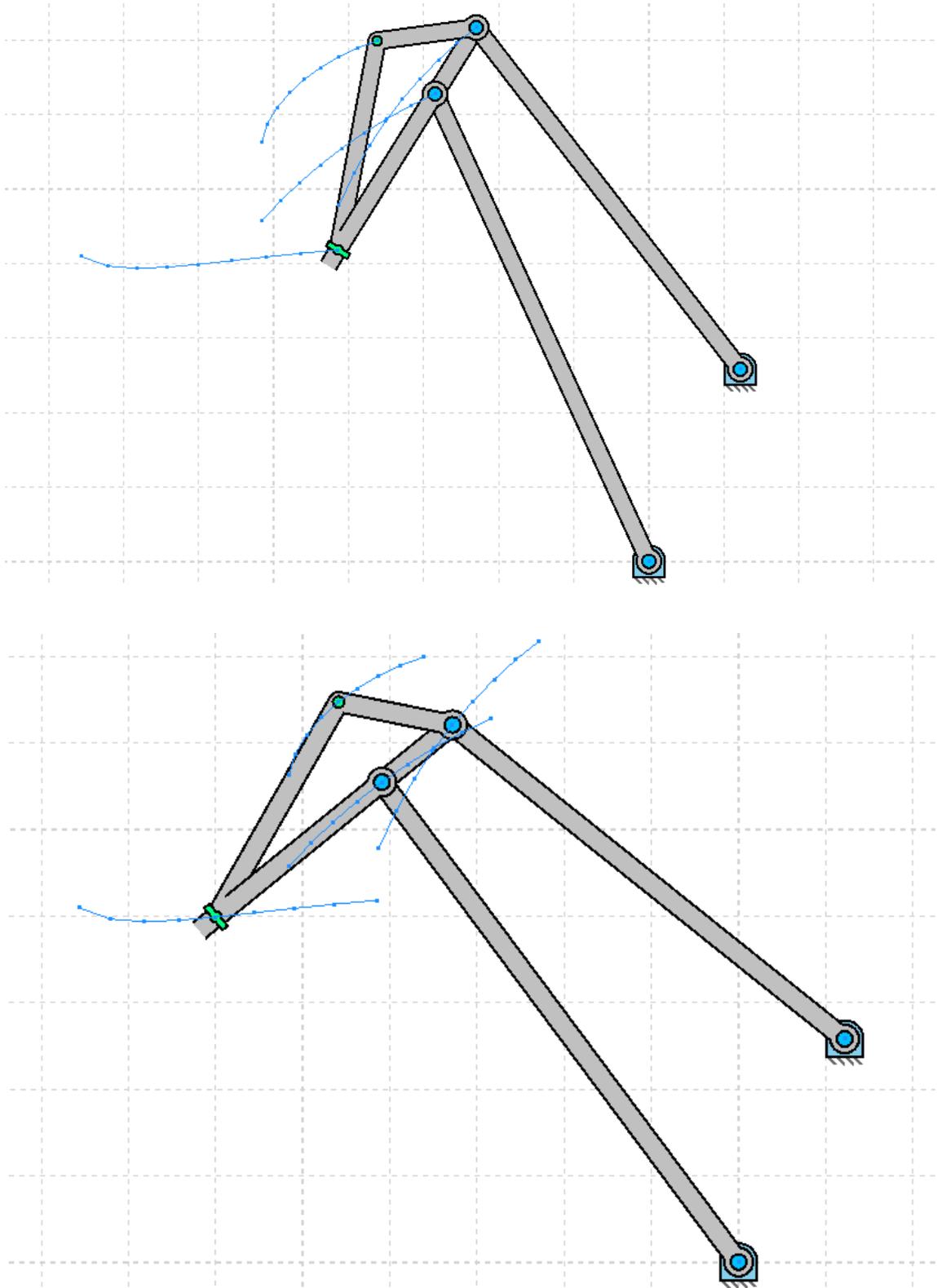


ILUSTRACIÓN 11: DISEÑO CUADRILÁTERO ARTICULADO

2.3.5.-Sistema de recogida del tirante

El sistema de recogida del tirante va situado sobre la base de la maqueta debido a la complejidad que supondría situar el motor sobre la plataforma giratoria por su gran tamaño. De este modo, el motor va situado sobre la base y en él se acopla un eje donde se recogerá el cable. Además, dicho eje tendrá una pieza como tope para que el tirante no se escape por su extremo.

2.4.-Construcción de la maqueta

En primer lugar, se han realizado cuatro taladros en la base de madera, tal y como se muestra previamente en el diseño (fig 8).

Una vez realizados los taladros, se ha construido el pórtico y fijado sobre la base de madera.

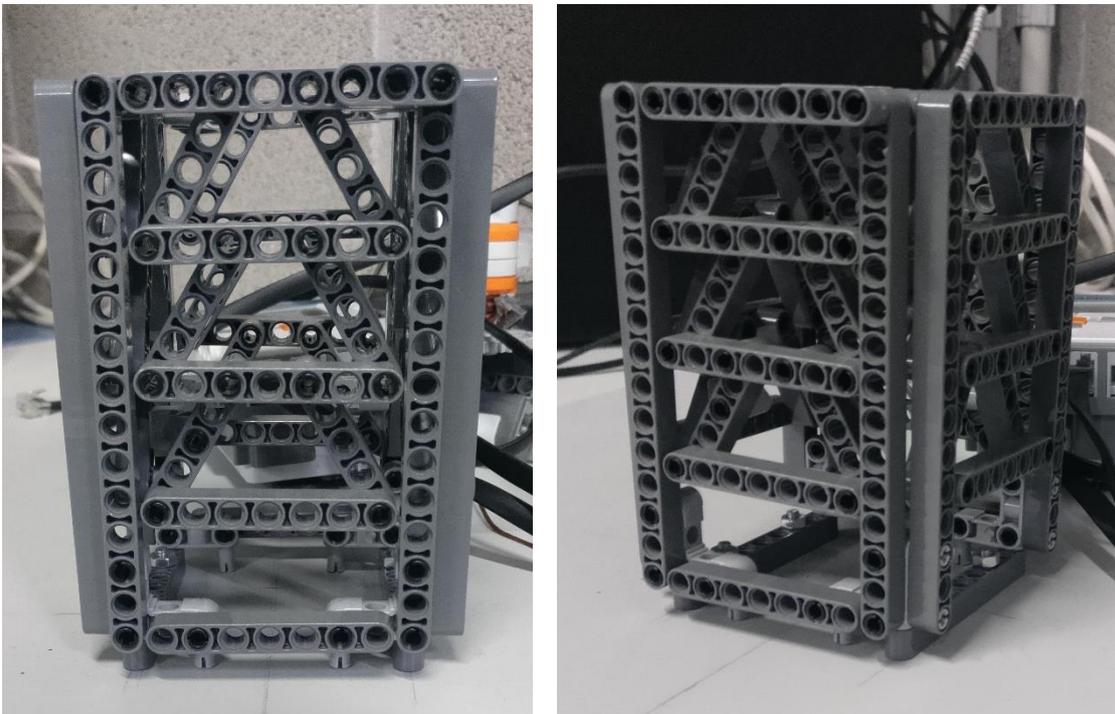


ILUSTRACIÓN 12: CONSTRUCCIÓN PÓRTICO

Después, hay que construir el mecanismo de la plataforma giratoria, y hacer la unión con el pórtico. Además, los sensores de luz irán situados en la parte fija de la plataforma giratoria como se puede observar en las siguientes imágenes.

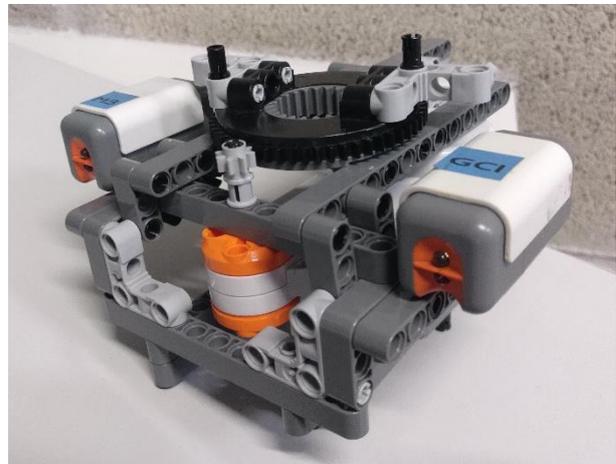


ILUSTRACIÓN 13: CONSTRUCCIÓN PLATAFORMA GIRATORIA

Luego, se realiza la construcción del cuadrilátero articulado con la ayuda del mecanismo calculado mediante GIM. También se realiza la unión entre dicho cuadrilátero y la plataforma giratoria.

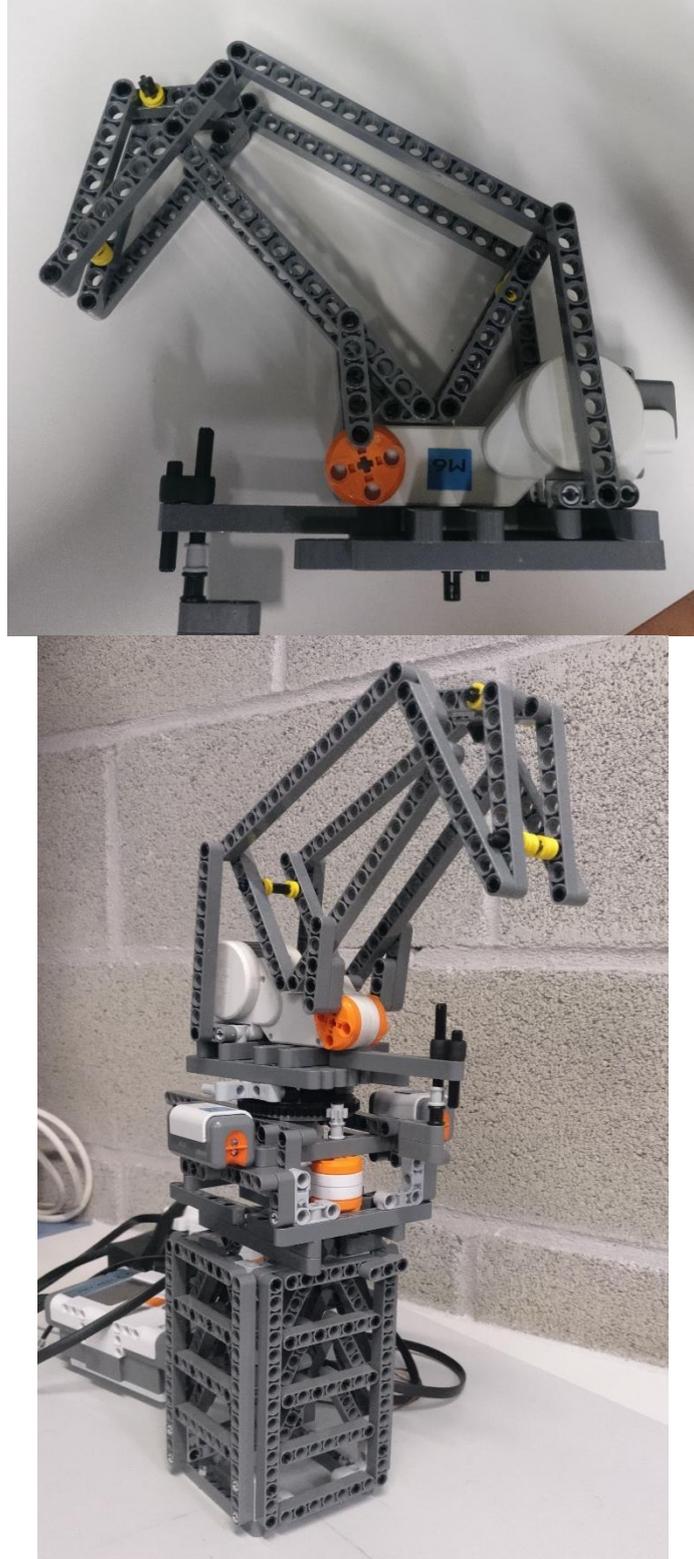


ILUSTRACIÓN 14: CONSTRUCCIÓN CADRILÁTERO ARTICULADO

2.5.-Puesta en marcha del entorno de programación

Una vez terminada la construcción de la estructura de la maqueta, la siguiente tarea a realizar es el control del movimiento de la grúa con la ayuda del programa BricxCC y usando el lenguaje NXC.

El objetivo del control de la maqueta de la grúa portuaria es el de transportar un objeto desde una posición inicial, (por ejemplo, un barco), hasta una posición final, (por ejemplo, el muelle del puerto).

Para dotar de más realismo a la maqueta vamos a mover una pieza metálica desde un punto hasta otro usando un electroimán situado en la punta del tirante y que se activará de forma manual.

Diagrama de flujo

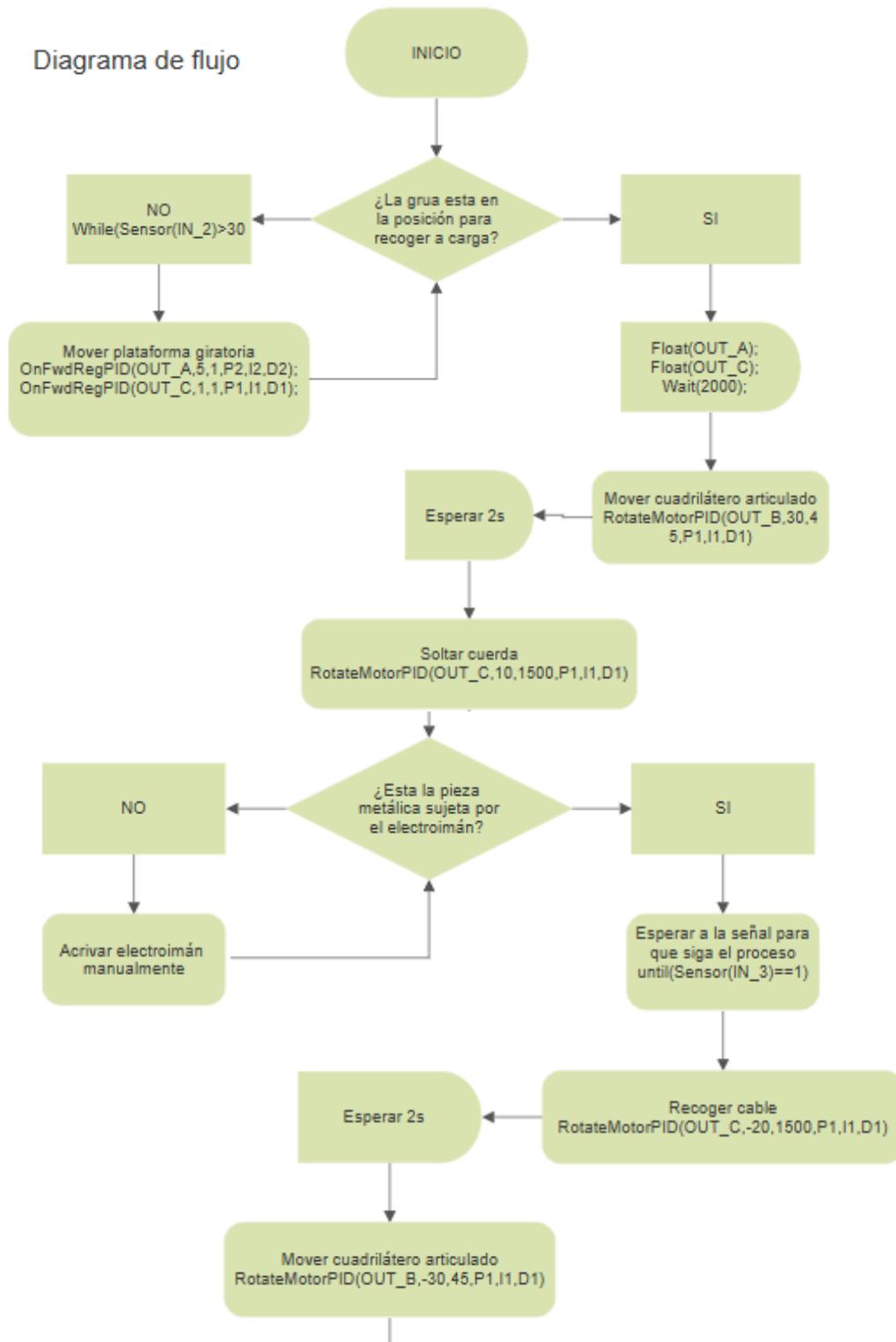


ILUSTRACIÓN 15: DIAGRAMA DE FLUJO CONTROL MOVIMIENTO

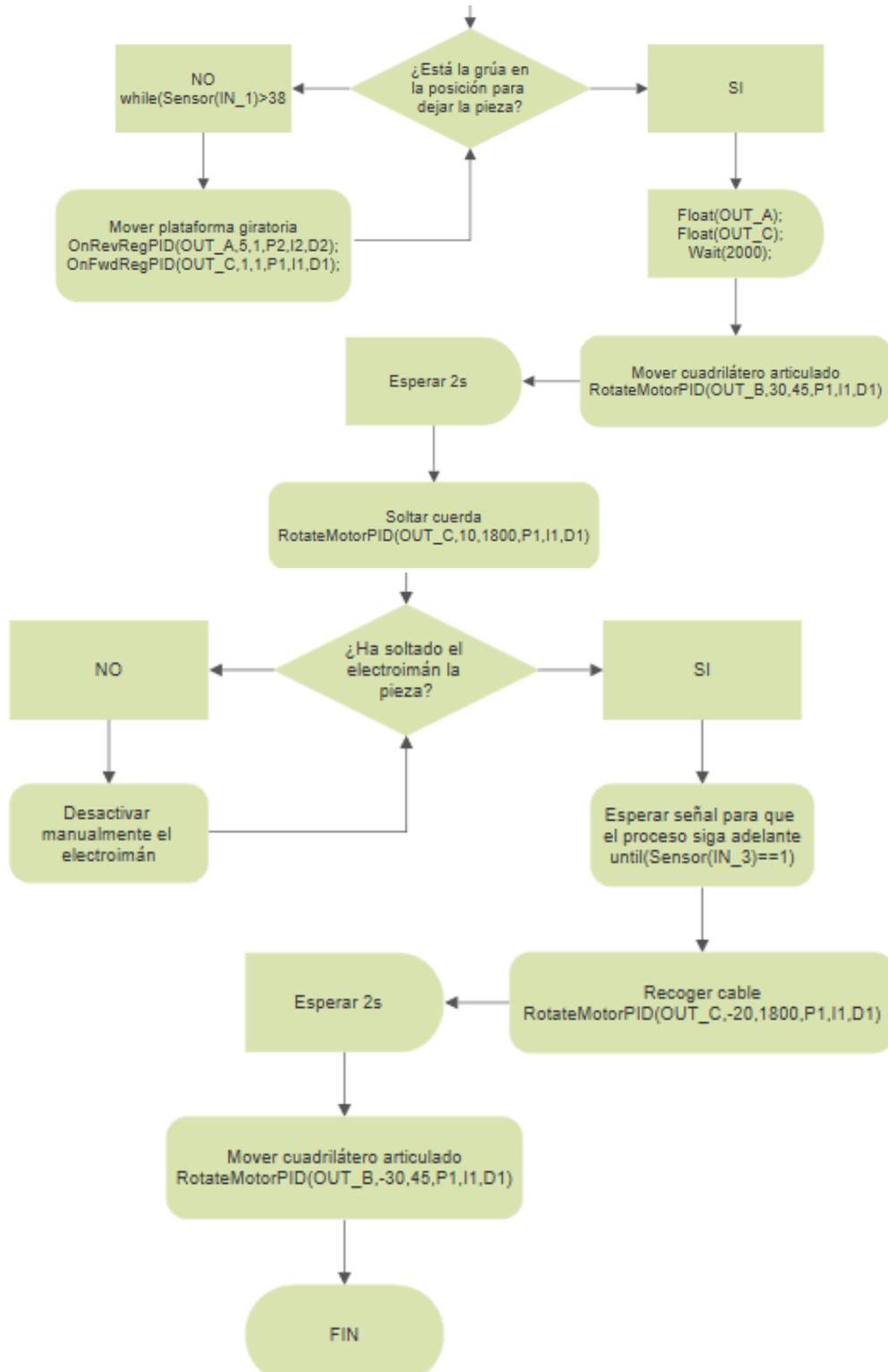


ILUSTRACIÓN 16: DIAGRAMA DE FLUJO CONTROL MOVIMIENTO

Al poner en marcha el programa la grúa tiene la orden de moverse hasta la posición en la que tiene que recoger la pieza metálica. Para ello, el valor de un sensor de luz, (IN_2) indicará cuando la estructura ha llegado a su posición, puesto que está disminuyendo la luminosidad en torno al sensor y este mandará una señal al programa. Para que este proceso se ejecute de manera correcta el programa realizará una calibración del sensor. Para ello, antes de realizar cualquier movimiento se guardará el valor de los dos sensores en dos variables que se han declarado previamente y servirán como referencia para que cuando el sensor lea un valor inferior al de dichas variables la grúa se detenga. Como se puede observar, a la vez que se está realizando el movimiento de la plataforma giratoria alrededor de su eje vertical (OUT_A), también se ha dado la orden de que el motor que controla la longitud del cable (OUT_C) lo suelte poco a poco. La razón por la que se realiza esta operación es evitar que el cable dificulte el giro de la plataforma, cosa que puede suceder debido a que el motor que controla la longitud del cable está situado sobre la base de la maqueta y no sobre la plataforma giratoria, por lo tanto, la distancia respecto a la plataforma no va a ser siempre la misma debido a su movimiento.

Una vez la grúa esté situada en la posición adecuada, es decir, sobre la pieza metálica que se va a transportar, el cuadrilátero articulado se moverá haciendo que la punta del pescante avance hacia delante. Asimismo, el motor que controla la longitud del cable se pondrá en marcha con el objetivo de que el electroimán, situado en el extremo del cable, se sitúe sobre la pieza metálica. Una vez esto haya sucedido, se activará el electroimán para que la pieza se adhiera a él. Una vez que la pieza esté correctamente sujeta por el electroimán, se dará la orden al sistema para que siga adelante. Esta acción se realiza mediante un sensor de contacto, activado manualmente.

A partir de este punto, el objetivo es llevar la pieza recogida hasta otra ubicación. Para ello, se recogerá el cable y se hará que el cuadrilátero articulado vuelva a su posición inicial, para posteriormente realizar el giro de la plataforma sobre su eje vertical. El giro se realiza de la misma forma, pero en sentido contrario y mediante otro sensor diferente, para indicarnos que se ha llegado a la posición en la que se quiere depositar la pieza. De esta manera, y en el momento en que la plataforma giratoria haya llegado a dicha posición, se volverá a realizar tanto el movimiento del cuadrilátero articulado como el del motor que controla la longitud del cable. Cuando la pieza se encuentre en la posición correcta, de manera manual, se apagará la fuente de alimentación del electroimán, el flujo magnético se detendrá y la pieza se soltará del electroimán.

Finalmente, sabiendo que la pieza está en su posición final y que ya no se encuentra adherida al electroimán el programa ordenará los últimos movimientos mediante el sensor de contacto, activado manualmente. A partir de ahí, el cable será recogido y el cuadrilátero volverá a su posición inicial, terminando de esta manera de realizar su cometido.

2.6.-Diseño del sistema de regulación de velocidad

Uno de nuestros objetivos principales es que el movimiento de nuestra maqueta se asemeje lo máximo posible al de la grúa real. No obstante, nos encontramos con un problema. A la hora de programar el movimiento de la maqueta no se han tenido en cuenta los efectos de fricción que surgen por motivo del peso de los elementos, así como de que la transmisión del movimiento mediante los engranajes que constituyen el mecanismo que realiza el movimiento de la plataforma giratoria sobre su eje vertical no es perfecta. Por ello, la velocidad de giro de la plataforma no es constante, es decir, su valor oscila a lo largo de su trayectoria debido a los efectos de la fricción. Asimismo, no es posible obtener el valor de velocidad deseado.

Con el fin de controlar esa velocidad y conseguir que su valor sea el deseado se va a diseñar un sistema de control realimentado que regule la velocidad de giro. Antes de nada, es necesario conocer el comportamiento de nuestro sistema, es decir, saber cual es su función de transferencia.

2.6.1.-Obtención del modelo de la planta

Una vez obtenidos los datos de velocidad y del tiempo mediante el programa BixCC se descargan en el ordenador y con el uso del Excel se creará un gráfico de la velocidad del motor respecto del tiempo.

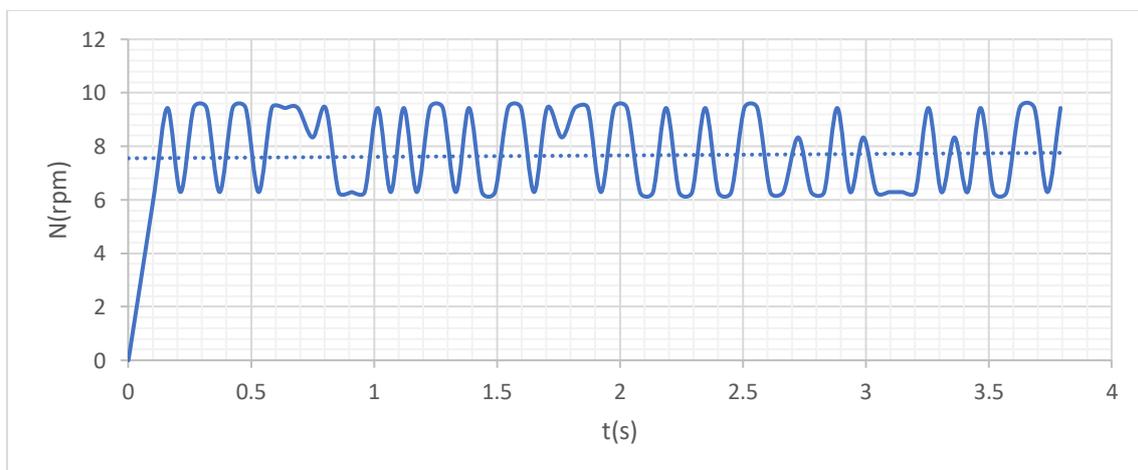


ILUSTRACIÓN 17: GRÁFICO N_T SIN CONTROLADOR

Con los datos obtenidos creamos un gráfico que muestra como varía la velocidad de giro de nuestro motor a lo largo del tiempo. Todo esto con una orden de que gire a 10rpm.

Como se puede observar en el gráfico, con una referencia de 10rpm obtenemos unos valores que oscilan entre las 6rpm y 9rpm. Para obtener la función de transferencia del sistema es necesario conocer cuál es el tipo de sistema a calcular. Las oscilaciones que presenta el gráfico son debidas a las holguras entre las piezas de plástico que se usan y sobre todo al ruido introducido por los engranajes internos del servomotor, que son de material plástico y poco robustos. Por lo tanto, teniendo en cuenta esa consideración y tomando un valor medio como valor de régimen permanente, se puede decir que el sistema se asemeja a uno de primer orden.

A continuación, se presenta la función de transferencia de un sistema de primer orden:

$$G(s) = \frac{K}{\tau s + 1} \left[\frac{rpm}{\% potencia} \right]$$

La función de transferencia a calcular es $G(s)$, la cual consta de dos parámetros:

- **K**: ganancia, se calcula dividiendo el valor estacionario de nuestra salida entre el valor del escalón unitario que se ha usado como entrada del sistema.

$$K = \frac{Y_{ss}}{a_0} = \frac{7,6815}{10} = 0,76815$$

- **τ** : Constante de tiempo, es el tiempo que tarda el sistema en llegar al 63% de su valor estacionario.

$$0,63 * Y_{ss} = 4,9527 \rightarrow \tau = 0,1s$$

Por lo tanto, la función de transferencia de nuestro sistema es la siguiente:

$$G(s) = \frac{0,76815}{0,1s + 1}$$

La función de transferencia de la planta corresponde al sistema formado por el conjunto estructura-motor-encoder integrado en la maqueta.

2.6.2.-Sintonización del controlador PID

El objetivo del diseño del controlador es que el sistema consiga alcanzar la velocidad de giro de 10rpm sin error.

Como se puede observar en la respuesta, en lazo abierto, no es capaz de llegar a las 10rpm. Por ello, se decide implementar un sistema de control realimentado.

El sistema con el que trabajamos es de tipo 0 y la entrada un escalón. Por ello, existe un error de régimen permanente. Para eliminar el error de régimen permanente ante una entrada escalón es necesario que el sistema sea al menos de tipo 1, es decir, es necesario añadir acción integral. En consecuencia, se decide implantar un controlador PI. Se desecha la opción de diseñar un controlador PID debido a los problemas de ruido ya comentados y que la acción derivativa podría contribuir a aumentar.

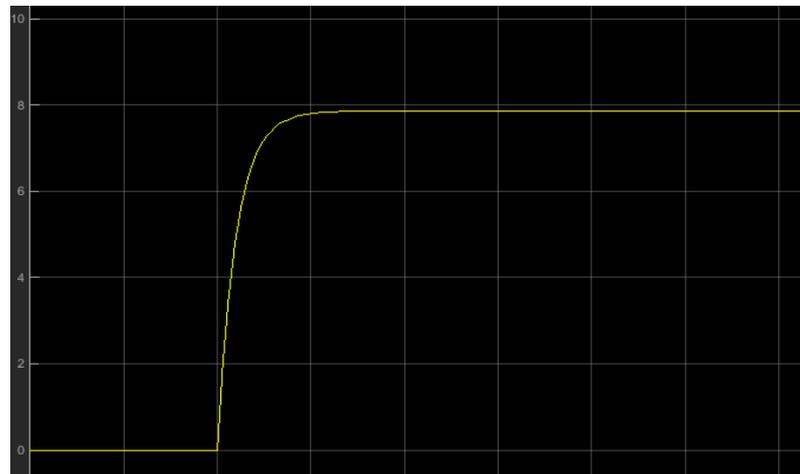
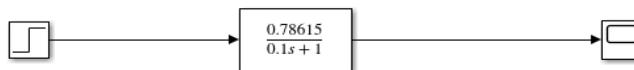


ILUSTRACIÓN 18: SIMULACIÓN SISTEMA SIN CONTROLADOR

Una vez conocido todo esto, se va a realizar un lazo realimentado mediante Simuink y se va a sintonizar un controlador PI para eliminar el error de régimen estacionario.

En la ilustración 19 se muestra el diagrama del sistema realimentado con el controlador PI implementado.

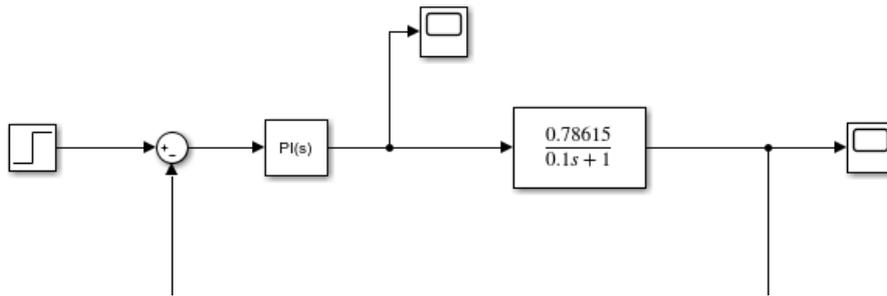


ILUSTRACIÓN 19: SISTEMA DE CONTROL REALIMENTADO

También se ha buscado que el sistema fuera lo más rápido posible, teniendo en cuenta que la acción de control sea razonable, de modo que el actuador no quede saturado.

En la siguiente imagen se puede ver el valor calculado para nuestros parámetros.

Controller parameters

Source:	internal	Compensator formula
Proportional (P):	3.06573067420986	$P + I \frac{1}{s}$
Integral (I):	44.1599767608784	
Select Tuning Method:	Transfer Function Based (PID Tuner App)	Tune...

ILUSTRACIÓN 20: CONTROLADOR PI

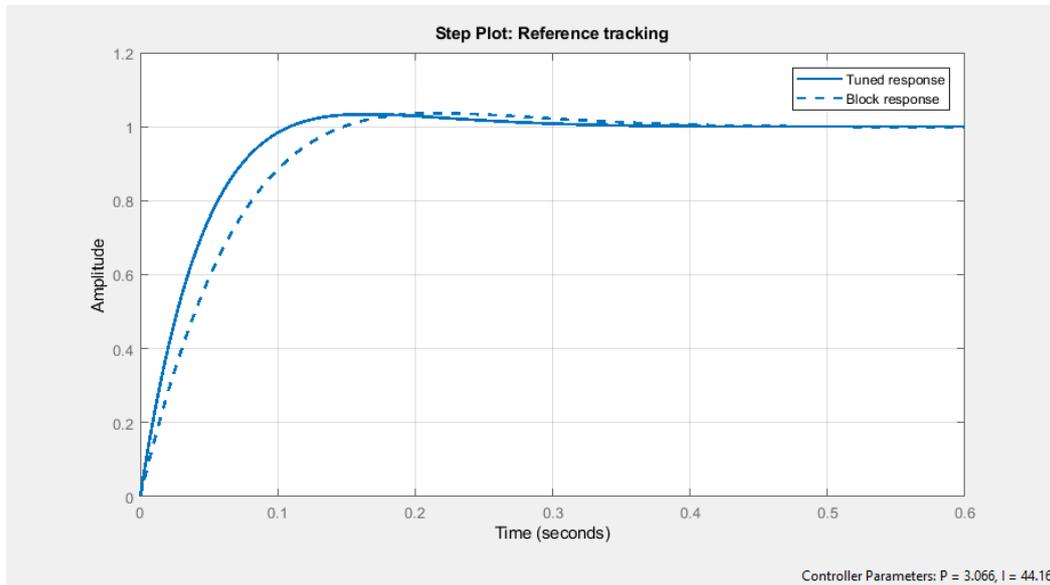


ILUSTRACIÓN 21: SINTONIZACIÓN CONTROLADOR

Además, es necesario verificar la señal de control puesto que nuestro sistema tiene unas limitaciones. Solo admite señales entre los valores de 0 y +100. Se puede ver que los valores están dentro de ese rango, por lo que el diseño del controlador es válido. Otra forma de solucionar el problema sin necesidad de realizar este paso hubiera sido añadir un bloque de saturación a la entrada de nuestro sistema.

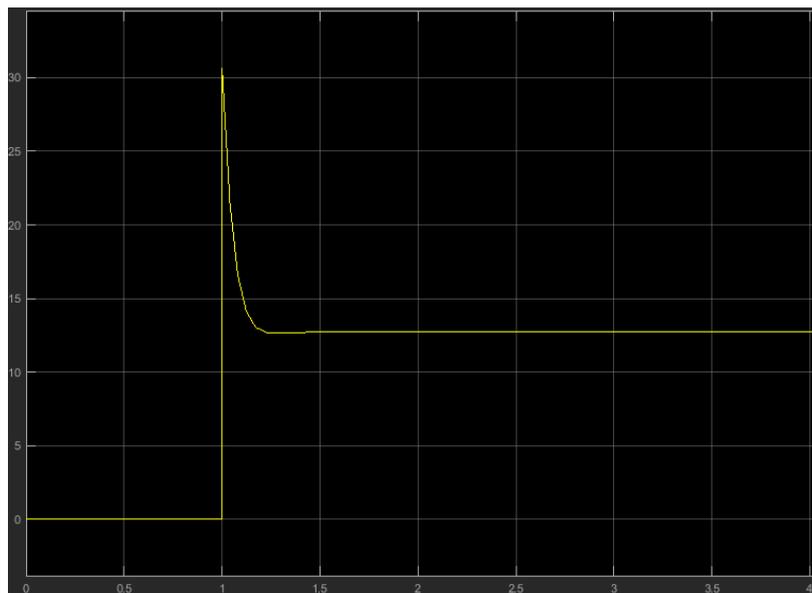


ILUSTRACIÓN 22: SEÑAL DE CONTROL

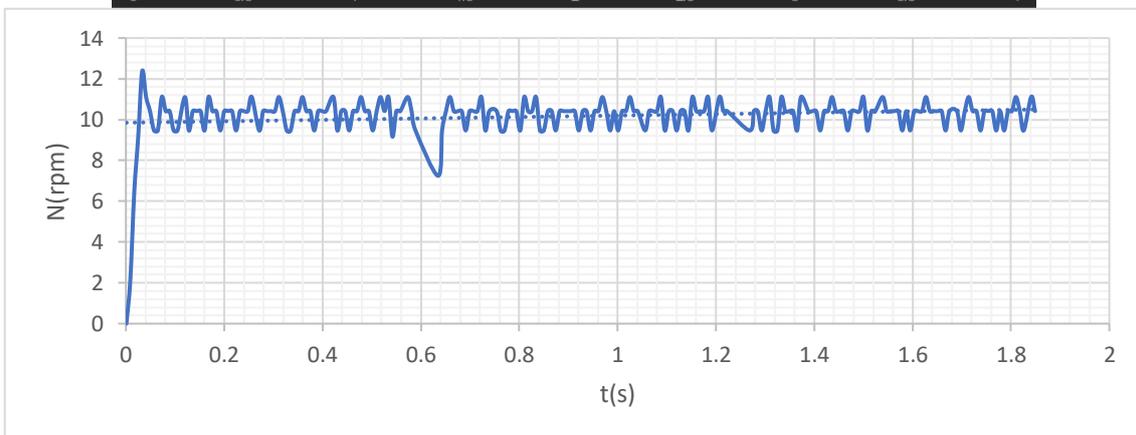
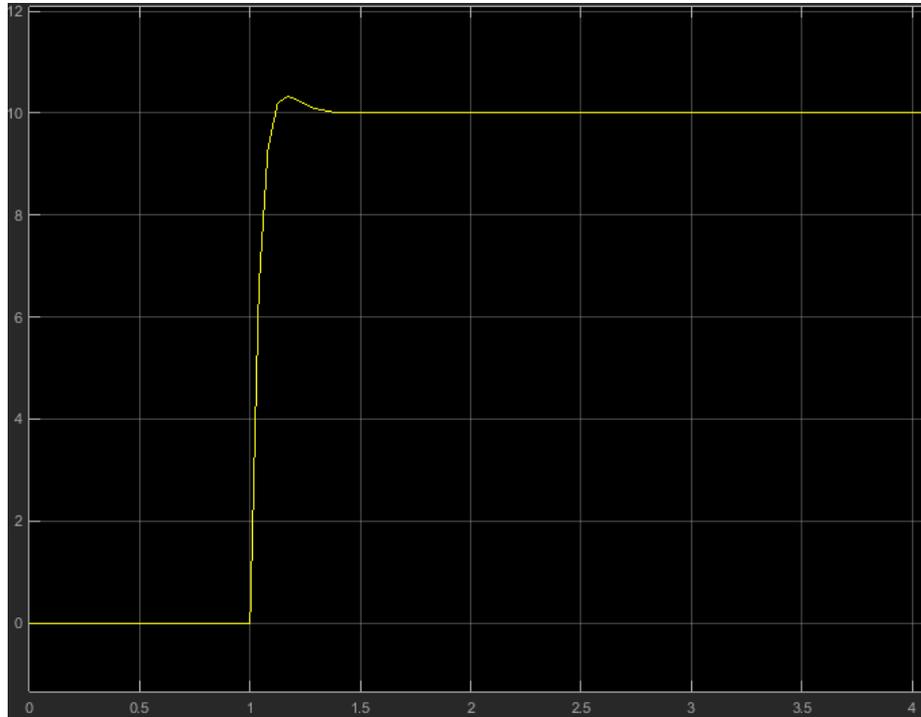


ILUSTRACIÓN 23: RESPUESTA DEL CONTROLADOR SIMULADO Y REAL CON CONTROLADOR PI

Finalmente, se pueden observar la respuesta del sistema, así como la respuesta de la maqueta una vez regulada su velocidad a 10 rpm.

Se puede observar como la respuesta del sistema calculada mediante Simulink y la respuesta del sistema obtenida mediante métodos experimentales son muy parecidas. Asimismo, el error de régimen estacionario ha sido eliminado y el valor del sobre impulso no es muy grande. Por lo tanto, se puede decir que el controlador calculado es aceptable y será el que se implemente en el control de la maqueta.

Por otro lado, se va a realizar el cálculo del controlador de manera analítica y se va a realizar una comparación con el calculado mediante Matlab y Simulink.

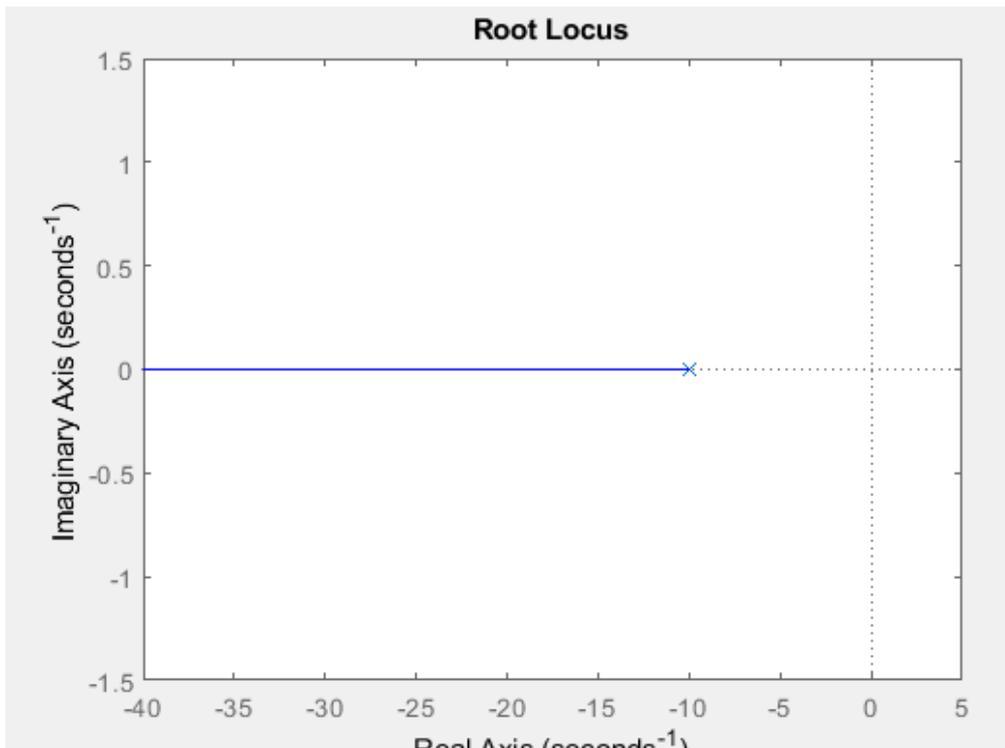


ILUSTRACIÓN 24: LUGAR DE RAICES PLANTA

En un principio este es el lugar de las raíces del sistema en lazo cerrado. La estrategia de diseño es introducir un controlador PI, anulando el polo de la planta con el cero que introduce el controlador.

$$G_c(s) = K_p \left(1 + \frac{1}{T_i * s} \right)$$

Esta es la función de transferencia del controlador. Primero se escogerá el valor de T_i , que anulará el polo de la planta:

$$T_i = 0,1s$$

Una vez escogido el valor de T_i , la función de transferencia del controlador más la planta es la siguiente:

$$G(s) = \frac{K_p * 7,6815}{0,1 * s}$$

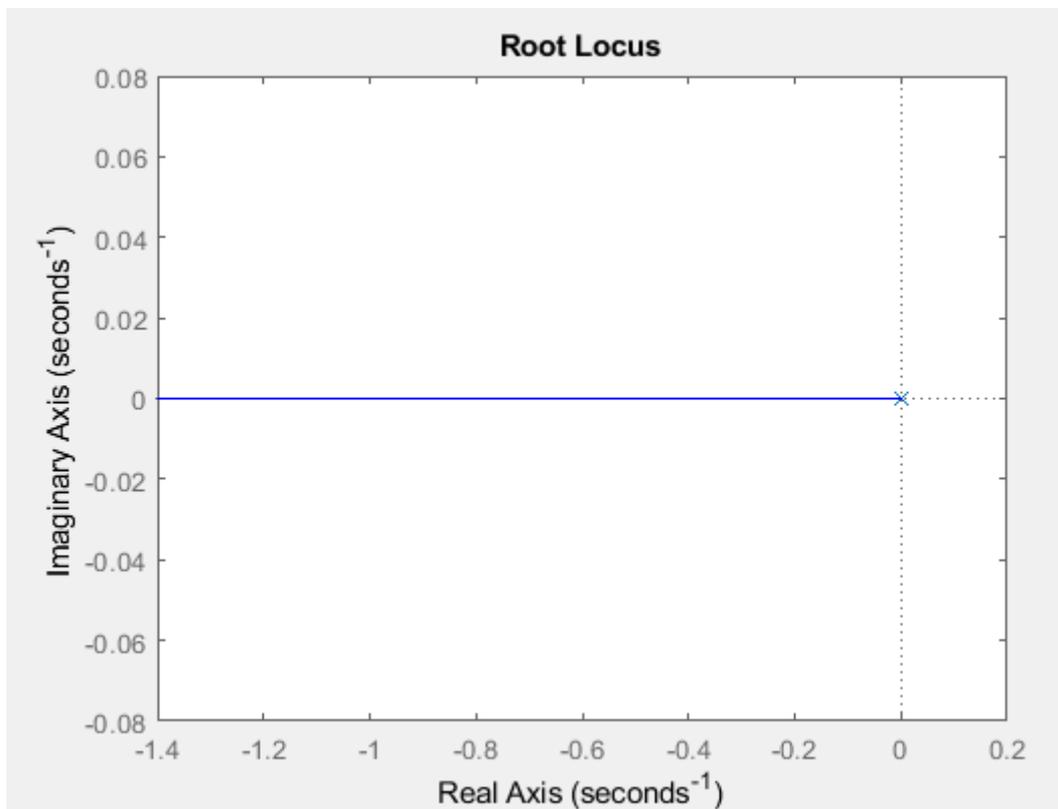


ILUSTRACIÓN 25: LUGAR DE LAS RAICES PLANTA+CONTROLADOR

En la figura 25 se muestra el lugar de las raíces del sistema en lazo cerrado una vez introducido el controlador PI. Falta calcular el valor de K_p , con el cual situaremos el polo de nuestro sistema en el lugar que nos convenga. De esta manera, cuanto mayor sea el valor de K_p más rápido será el sistema, aunque la señal de control también aumentará y eso podría crear un riesgo de saturación del motor que debe evitarse.

A continuación, se muestra el valor de los parámetros escogidos para el controlador:

Controller parameters

Source: [Compensator formula](#)

Proportional (P):

Integral (I):

Select Tuning Method:

$P + I \frac{1}{s}$

ILUSTRACIÓN 26: CONTROLADOR PI

Finalmente, se van a mostrar la señal de control, la señal de salida simulada mediante Simulink y la obtenida experimentalmente. Asimismo, estas señales se van a comparar con las obtenidas con el controlador sintonizado mediante el Autotune.

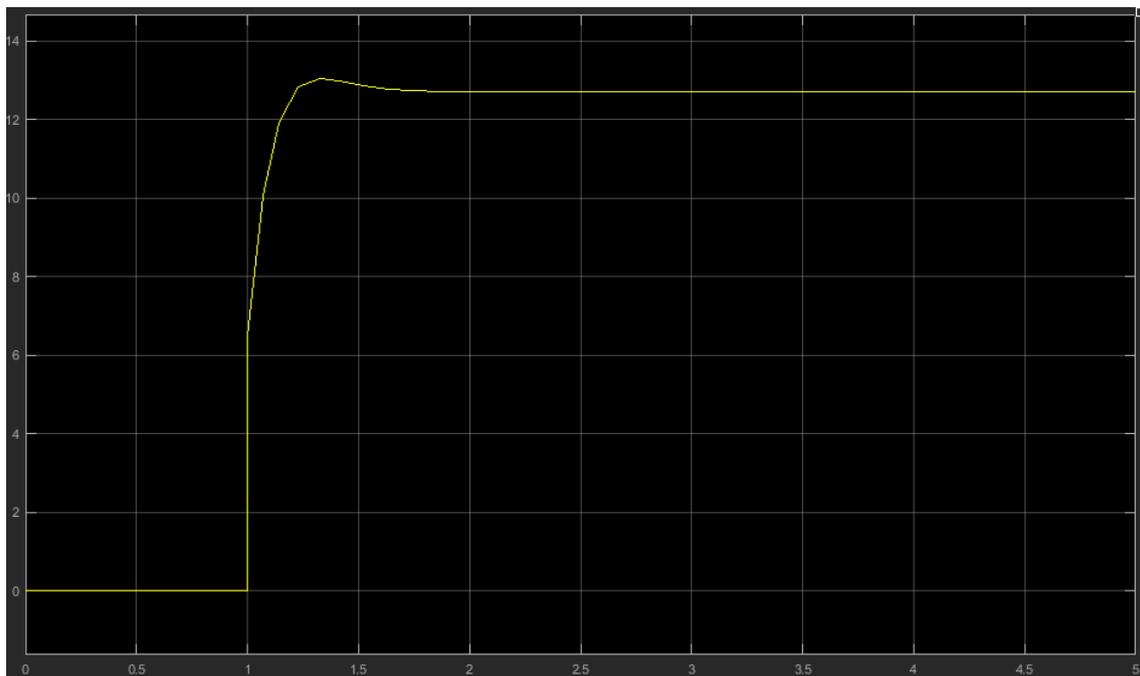


ILUSTRACIÓN 27: SEÑAL DE CONTROL

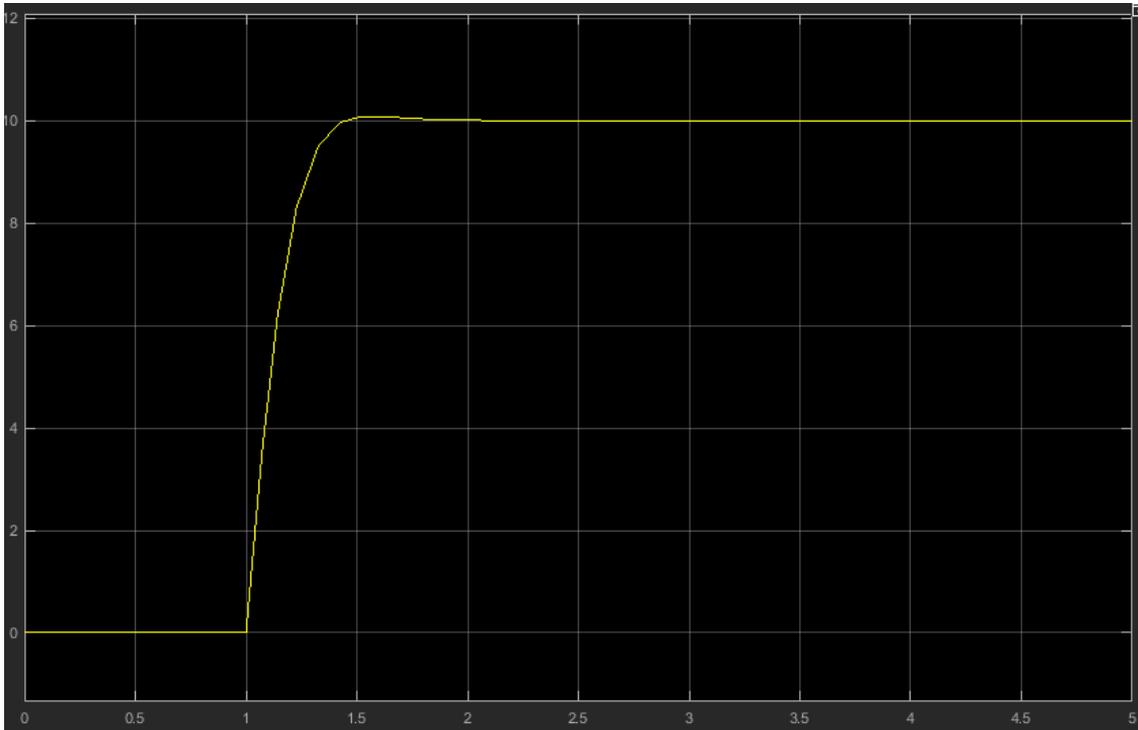


ILUSTRACIÓN 28: SEÑAL SALIDA SIMULADA

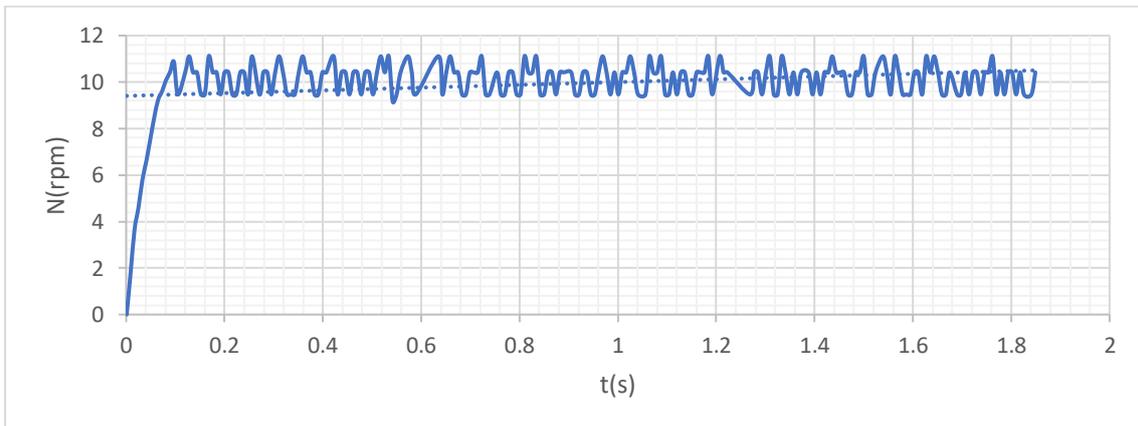


ILUSTRACIÓN 29: SEÑAL DE SALIDA EXPERIMENTAL

Como podemos observar, la salida del sistema tiene un sobreimpulso pequeño y es casi idéntica a la respuesta de un sistema de primer orden. Además, se ha conseguido eliminar el error de régimen permanente. En cuanto a la señal de control, no hay saturación. La respuesta obtenida experimentalmente es parecida a la de la simulación, aunque como nos ha ocurrido anteriormente, se pueden observar los efectos del ruido en la señal en el sistema real.

Por último, hay que realizar la comparación del controlador realizado analíticamente con el calculado con Autotune:

- La señal de control del controlador realizado analíticamente no obtiene unos valores tan altos como la otra. No hay saturación en ninguno de los dos casos.

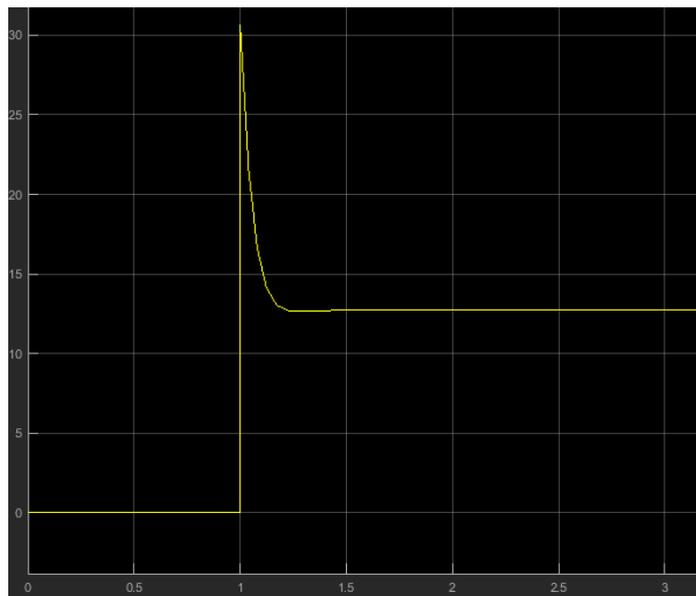


ILUSTRACIÓN 30: SEÑAL CONTROL "AUTOTUNE"

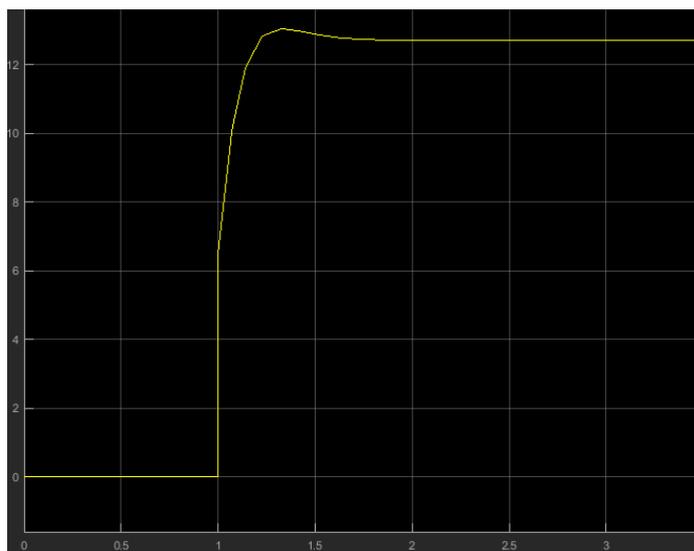


ILUSTRACIÓN 31: SEÑAL CONTROL ANALÍTICO

- La salida del controlador calculado mediante Simulink es más rápida que la calculada analíticamente. No obstante, tiene un pequeño sobreimpulso, que la salida del controlador analítico no presenta.

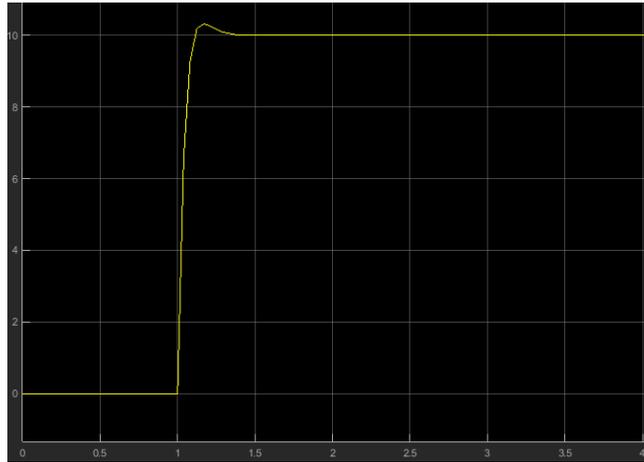


ILUSTRACIÓN 32: SALIDA "AUTOTUNE"

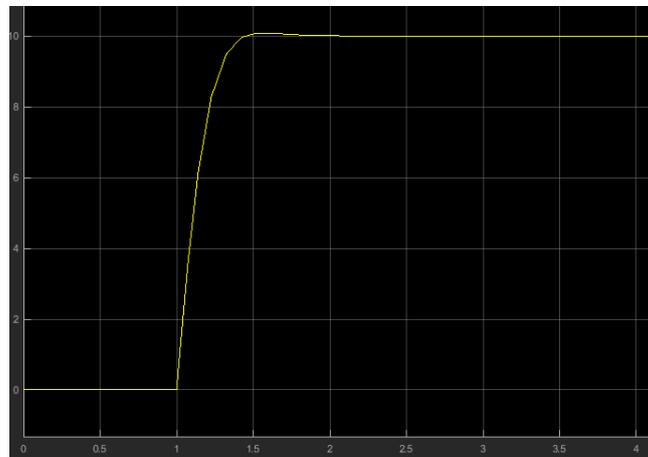


ILUSTRACIÓN 33: SALIDA ANALÍTICO

Por estas dos razones, se puede decir que los dos controladores son aceptables y cumplen las especificaciones que se piden. Aun así, si se quisiera usar el controlador más rápido se tendría que elegir el calculado con el Autotune.

3.-Aspectos económicos

3.1.-Mano de obra

HORAS INTERNAS	Coste hora (€/h)	Horas trabajadas (h)	Coste total (€)
Ingeniero	20,00	180	3600,00
			3600,00

TABLA 3: MANO DE OBRA

3.2.-Gastos de montaje

PRODUCTO	Coste(€)
Tabla de madera 550x420x15	9,99
Pack 8 tornillos cabeza cilíndrica allen M4x6	1,99
Pack 8 tuercas M4	1,99
2 de 12V 0,24 A 2,5 kg imán eléctrico elevación electroimán solenoide 20x15mm	6,92
Cuerda 1m	1,99
VELCRO VEL-EC60241, Cinta adhesiva extra fuerte, 50mm x 1m, Negro	10,04
39,92	

TABLA 4: GASTOS DE MONTAJE

3.3.-Gastos de Lego Mindstorms

PRODUCTO	Coste(€)
Kit Lego Mindstorms NXT	399,99
399,99	

TABLA 5: GASTOS DE LEGO MINDSTORMS

3.4.-Gastos de Software

PRODUCTO	Coste(€)
Licencia Microsoft Office 365 Hogar	99,00
MATLAB and Simulink Student Suite R2019a	82,00
	181,00

TABLA 6: GASTOS DE SOFTWARE

3.5.-Presupuesto ejecutado

PRESUPUESTO EJECUTADO(€)	
Mano de obra	3600,00
Material montaje	39,92
Lego Mindstorms	399,99
Software	181,00
Subtotal	4220,91
Imprevistos(10%)	422,09
TOTAL	4643,00

TABLA 7: PRESUPUESTO EJECUTADO

4.-Conclusiones

Tras la finalización del proyecto y observar el funcionamiento de la maqueta de la grúa portuaria se da por cumplido el objetivo inicial de realizar la construcción y el control de esta.

La maqueta que ha sido construida tiene un gran parecido con la grúa que se ha tomado como referencia, lo cual era una premisa que teníamos desde un principio, aunque debido a las limitaciones del material usado tenga cierta diferencia.

Además, se ha conseguido crear un programa capaz de controlar el movimiento de la maqueta, el cual aporta realismo a la misma. Asimismo, se ha realizado un estudio para mejorar el movimiento de la maqueta. Para ello se han identificado cuales son los aspectos principales que mejorar y se decide que el más relevante es el movimiento de la plataforma giratoria de la maqueta, por lo cual, se ha decidido implementar un controlador que permita un funcionamiento más preciso.

Para la realización de esta tarea, los datos de velocidad han sido obtenidos mediante experimentación y se han ilustrado en una gráfica, con la cual se ha calculado el modelo del sistema a controlar para poder implantar un controlador PI.

Finalmente, para el cálculo del controlador se ha usado el programa Matlab, y esto nos permite simular el sistema obtenido previamente con la acción del controlador y poder obtener los valores del mismo que logran que el sistema cumpla las especificaciones que se desean. También se ha calculado un controlador de forma analítica, con ayuda de métodos aprendidos en el grado, y se ha comparado con el controlador anterior.

5.-Bibliografía

- [1] *“The LEGO MINDSTORMS EV3 Idea Book: 181 Simple Machines and Clever Contraptions”*. (s.f.).
- [2] *BricxCommandCenter*. (s.f.). Obtenido de <http://bricxcc.sourceforge.net/>
- [3] *NXC*. (s.f.). Obtenido de <http://blog.electricbricks.com/2009/10/nxc/>
- [4] *Programmable robots, robotics kit, electronics, education robots - Génération Robots. NXT-G: the development environment supplied with Lego Mindstorms, NXT-G*. (s.f.). Obtenido de <http://www.generationrobots.com/blog/en/nxt-g-the-development-environment-supplied-with-lego-mindstorms-nxt-g/>
- [5] *programming., Home of the best robot programming language for Educational Robotics. Made for NXT programming and VEX*. (s.f.). Obtenido de <http://www.robotc.net/download/>
- [6] *Wikipedia, enciclopedia libre*. (s.f.). Obtenido de http://es.wikipedia.org/wiki/Lego_Mindstorms
- [7] *Wikipedia, la enciclopedia libre*. (s.f.). Obtenido de http://es.wikipedia.org/wiki/Itsasmuseum_Bilbao

Anexo I: Manual de usuario

Antes de ejecutar el programa es indispensable asegurarse de que cada motor y sensor está conectado con el puerto que le corresponde. El motor que se encarga del movimiento de la plataforma giratoria tiene que ir conectado al puerto A, el motor que realiza el movimiento del cuadrilátero articulado al puerto B y el que recoge el cable al C. En cuanto a los sensores, el sensor más cercano al ladrillo se conecta al puerto 2 y el más lejano al puerto 1. Además, dispondremos de un sensor de contacto conectado al puerto 3.

Una vez hemos realizado esta tarea, hay que abrir el programa BricxCC y abrir el archivo que realiza el movimiento de la maqueta.

Una vez abierto el archivo, lo compilamos y ya estaría preparado para ser ejecutado.

Finalmente, se ejecutará el programa y la maqueta empezará a moverse. Una vez el cable haya bajado habrá que activar el electroimán manualmente y dar la señal al sensor 3 para que el programa siga funcionando. Habrá que hacer lo mismo cuando el cable vuelva a bajar para soltar la pieza. Una vez la pieza a sido soltada se vuelve a dar la señal al programa mediante el sensor de contacto que se activa manualmente para que la grúa vuelva a su posición inicial.

Anexo II: Manual de programación

```
#define P1 100
#define I1 20
#define D1 60
#define P2 3
#define I2 44
#define D2 0

task main()
{
int irSensorValue1; // valor leído por el sensor de luz 1
int irSensorValue2; // valor leído por el sensor de luz 2
SetSensorLight(IN_1); //definir sensor
SetSensorLight(IN_2); //definir sensor
SetSensor(IN_3,SENSOR_TOUCH); //definir sensor

irSensorValue1 = Sensor(IN_1)-2;
irSensorValue2 = Sensor(IN_2)-2;

while(Sensor(IN_2)>irSensorValue2 ){
OnFwdRegPID(OUT_A,10,1,P2,I2,D2); //motor en marcha
OnFwdRegPID(OUT_C,1,1,P1,I1,D1);} //Soltar cuerda
Float(OUT_A);
Float(OUT_C); //parar suavemente
Wait(2000); //esperar 2 segundos
```

```
RotateMotorPID(OUT_B,30,45,P1,I1,D1); //Mover cuadrilatero
```

```
Wait(2000); //esperar 2 segundos
```

```
RotateMotorPID(OUT_C,10,1500,P1,I1,D1); //Soltar cuerda
```

```
until(Sensor(IN_3)==1); //esperar hasta señal
```

```
RotateMotorPID(OUT_C,-20,1500,P1,I1,D1); //Recoger cuerda
```

```
Wait(2000); //esperar 2 segundos
```

```
RotateMotorPID(OUT_B,-30,45,P1,I1,D1); //Mover cuadrilatero
```

```
Wait(2000); //esperar 2 segundos
```

```
while(Sensor(IN_1)>irSensorValue1 ){
```

```
OnRevRegPID(OUT_A,10,1,P2,I2,D2); //motor en marcha
```

```
OnFwdRegPID(OUT_C,1,1,P1,I1,D1);} //Soltar cuerda
```

```
Float(OUT_A);
```

```
Float(OUT_C); //parar suavemente
```

```
Wait(2000); //esperar 2 segundos
```

```
RotateMotorPID(OUT_B,30,45,P1,I1,D1); //Mover cuadrilatero
```

```
Wait(2000); //esperar 2 segundos
```

```
RotateMotorPID(OUT_C,10,1800,P1,I1,D1); //Soltar cuerda
```

```
until(Sensor(IN_3)==1); //esperar hasta señal
```

```
RotateMotorPID(OUT_C,-20,1800,P1,I1,D1); //Recoger cuerda
```

```
Wait(2000); //esperar 2 segundos*/
```

```
RotateMotorPID(OUT_B,-30,45,P1,I1,D1); //Mover cuadrilatero  
Wait(2000); //esperar 2 segundos  
}
```

Anexo III: Manual de instalación

Para conectar el dispositivo con BricxCC hay que seguir los siguientes pasos:

- Abrir el programa BricxCC en el ordenador y seleccionar el tipo de conexión, el tipo de ladrillo y el firmware. En este caso se elegirán usb, NXT, y Standard respectivamente.
- Abrir el programa encargado del movimiento de la maqueta.
- Conectar el ordenador con el ladrillo mediante usb.
- Compilar y descargar el programa en el ladrillo.
- Ejecutar el programa.