



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

Desarrollo de algoritmos de control de un robot móvil
en interiores

Autor: Belén Castellote López

Director: Jaime Boal Martín-Larrauri

Director: Javier García Aguilar

Madrid

Agosto de 2020

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
“Desarrollo de algoritmos de control de un robot móvil en interiores”
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2019/2020 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Belén Castellote López

Fecha 26/08/2020

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Jaime Boal Martín-Larrauri

Fecha: / /



Fdo.: Javier García Aguilar

Fecha: 26 / 08 / 2020



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

Desarrollo de algoritmos de control de un robot móvil
en interiores

Autor: Belén Castellote López

Director: Jaime Boal Martín-Larrauri

Director: Javier García Aguilar

Madrid

Agosto de 2020

Agradecimientos

Me gustaría agradecerles a ambos directores del ICAI, Jaime Boal y Javier García, por su trabajo, constancia y paciencia a la hora de ayudarme a desarrollar el proyecto, pues las condiciones en el año 2020 han sido bastante desfavorecidas.

Gracias a mis compañeros, profesores, familia y amigos que han estado conmigo durante estos cuatro años en la universidad, por cuidarme y estar siempre pendientes.

DESARROLLO DE ALGORITMOS DE CONTROL PARA LA NAVEGACIÓN DE UN ROBOT MÓVIL EN INTERIORES.

Autor: Castellote López, Belén.

Director: Boal Martín-Larrauri, Jaime.

Director: García Aguilar, Javier.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

El proyecto consiste en el estudio de un control PID como herramienta para buscar la máxima eficacia y agilidad en el desplazamiento del robot. Se utilizan simulaciones para comprobar la eficacia del control creado. Una vez logrado ese control, se estudia un control predictivo, con el objetivo de buscar un resultado más eficiente. Se trabajará con Matlab como herramienta de programación de los controles y V-REP como herramienta de simulación del robot.

Palabras clave: Robots móviles, simulación, modelado de la planta, control PID.

1. Introducción

La industria 4.0 hace referencia a una nueva forma de organización que ha sido posible gracias a las tecnologías de la información [1]. Este término es el mismo que ha dado lugar a otros como M2M, Big data o robótica avanzada, donde tienen cabida proyectos que envuelvan a la robótica móvil y al control de estos.

Desde hace unos años, son muchas las empresas que están decidiendo trabajar con robots móviles internos, para facilitar el trabajo, aumentar la productividad y perder menos dinero. Así es el ejemplo de Amazon con su robot Kiva, que gracias a su dinamismo y fuerza es capaz de transportar estanterías de lado a lado del almacén a la vez que aumenta el espacio de almacenamiento hasta un 40% de su capacidad [2].

Este ejemplo del robot de Amazon es sólo uno más de los fantásticos resultados que puede dejar ver la regulación automática. Ésta se encarga del diseño de controles para mejorar el comportamiento del robot.

2. Definición del proyecto

El proyecto realizado se divide en dos partes: el modelado de las plantas que tiene el robot Pioneer-P3DX con el que se trabaja; y el diseño del control de este mismo robot, que es simulado en V-REP con el objetivo de mandar y recibir datos para visualizar los resultados.

El robot puede variar sus sensores y características, por lo que, para esta simulación, puesto que se requiere que el robot sea capaz de seguir una pared, se le dota de sensores ultrasónicos con el objetivo de que pueda mantener cierta distancia de referencia.

3. Descripción del proceso

La primera parte del proyecto es la de vinculación entre las dos herramientas principales, que serían Matlab y V-REP, mediante una serie de comandos que se describen en la pagina de CoppeliaRobotics.

La segunda parte del proyecto es referente al modelado de la planta del robot, según sus tres formas de comportamiento a estudiar: navegación libre (velocidad de avance y velocidad de giro) y seguimiento a pared. Con estas plantas analizadas se procede a un posterior diseño de los controles. Las plantas obtenidas para cada uno han sido las siguientes:

$$P_{avance}(s) = \frac{1}{0.33 * s + 1} \quad P_{giro}(s) = \frac{1}{0.245 * s + 1}$$

$$P_{pared}(s) = \frac{-0.5 * 1 * \left(1 + \frac{-3.88e - 3}{0.5} * s\right)}{(0.245 * s + 1) * s^2}$$

En el diseño del control PID, se procede a la utilización de un programa de diseño de control PID para las plantas del robot, aunque previamente se ha estudiado según la forma de cada una de estas plantas cómo debe ser ese control.

En el diseño del control predictivo, se procede a la selección de las variables a tener en cuenta para el control y se trabaja con el programa de diseño *MPC Designer*.



Figura 1: Esquema del procedimiento a seguir en el proyecto

4. Resultados

Las plantas obtenidas para el comportamiento del robot en navegación libre son bastante sencillas, con un único polo y una ganancia. Por lo tanto, el diseño del control PID parece bastante sencillo, pues se deben estudiar exclusivamente las posibilidades de control P, PI y PID.

La planta de seguimiento a la pared es bastante más complicada, y debido a su forma con cero positivo, se decide proceder al diseño de un control en cascada, trabajando con dos controles diferentes: control P en lazo interno y control PD en lazo externo y viceversa. La Figura 2 muestra el resultado del control en cascada para PD en el interno y P en el externo. Con este último control, el tiempo que se obtiene al recorrer todo el circuito sin chocarse es de 1 minuto y 12 segundos.

K_{PD}	T_d	b	N	K_{ext}
-4.0816	0.245	1	10	3.24

Tabla 1: Parámetros control de seguimiento a pared óptimo

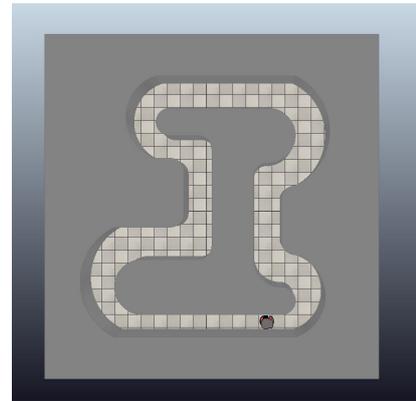
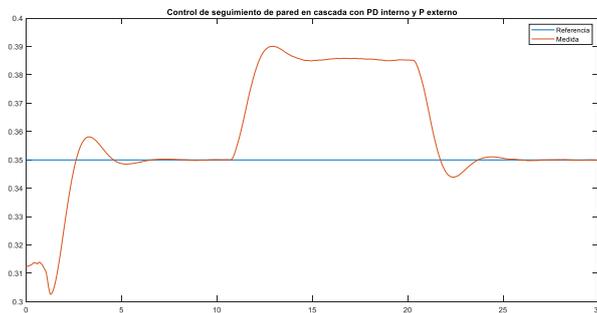


Figura 2: Control en cascada para planta de seguimiento a la pared y entorno de simulación

5. Conclusiones

A la hora de realizar la comparativa entre los diferentes controles PID que se han diseñado, se decide utilizar exclusivamente el control en cascada con control PD en el lazo interno y P en el externo, pues es el más eficaz. A la hora de intentar trabajar con control para la planta de velocidad de avance y para la planta de velocidad de giro a la vez que el control de la planta de seguimiento a la pared se observa que son de velocidades diferentes y que no se acompañan bien.

Puesto que no se han obtenido resultados en el control predictivo, se tiene en cuenta como un posible trabajo futuro, en el que se tengan en cuenta diferentes factores externos no medibles. Este proyecto sería realizable en un contexto académico.

6. Referencias

- [1] del Val Román, J. L. (2016). *Industria 4.0: la transformación digital de la industria*. In Valencia: Conferencia de Directores y Decanos de Ingeniería Informática, Informes CODDII. <http://coddii.org/wp-content/uploads/2016/10/Informe-CODDII-Industria-4.0.pdf> (Último acceso: 05/06/2020)
- [2] D'Andrea, R. (2012). Guest editorial: *A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead*. *IEEE Transactions on Automation Science and Engineering*, 9(4), 638-639. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6295687> (Último acceso: 19/08/2020)

DEVELOPMENT OF CONTROL ALGORITHMS FOR THE NAVIGATION OF A MOBILE ROBOT IN INDOORS.

Author: Castellote López, Belén.

Supervisor: Boal Martin-Larrauri, Jaime.

Supervisor: García Aguilar, Javier.

Collaborating Entity: ICAI – Universidad Pontificia Comillas

ABSTRACT

Keywords: Mobile robots, simulation, plant modeling, PID control.

1. Introduction

The project consists of the study of a PID control as a tool to seek maximum efficiency and agility in the movement of the robot. Simulations are used to check the effectiveness of the created control. Once this control is achieved, predictive control is studied, with the aim of seeking a more efficient result. We will work with Matlab as a control programming tool and V-REP as a robot simulation tool.

2. Project description

Industry 4.0 refers to a new form of organization that has been possible thanks to information technologies [1]. This term is the same that has given rise to others such as M2M, Big data, or advanced robotics, where there is room for projects that involve mobile robotics and their control.

For a few years, many companies are deciding to work with internal mobile robots, to facilitate work, increase productivity, and lose less money. This is the example of Amazon with its Kiva robot, which thanks to its dynamism and strength is capable of transporting shelves from side to side of the warehouse while increasing storage space up to 40% of its capacity [2].

This example of the Amazon robot is just one of the fantastic results that automatic regulation can show. This is responsible for the design of controls to improve the behavior of the robot.

3. Process description

The first part of the project is the link between the two main tools, which would be Matlab and V-REP, through a series of commands that are described on the CoppeliaRobotics page.

The second part of the project refers to the modeling of the robot plant, according to its three forms of behavior to study: free navigation (forward speed and raw speed) and wall tracking. With these analyzed plants, a subsequent design of the controls is carried out. The plant obtained were as follows:

$$P_{forward}(s) = \frac{1}{0.33 * s + 1} \quad P_{raw}(s) = \frac{1}{0.245 * s + 1}$$

$$P_{wall}(s) = \frac{-0.5 * 1 * \left(1 + \frac{-3.88e - 3}{0.5} * s\right)}{(0.245 * s + 1) * s^2}$$

In the design of the PID control, a PID control design program is used for the robot plants, although previously it has been studied, depending on the shape of each of these plants, how that control should be.

In the design of predictive control, the variables to be taken into account for the control are selected and the design program MPC Designer is used.

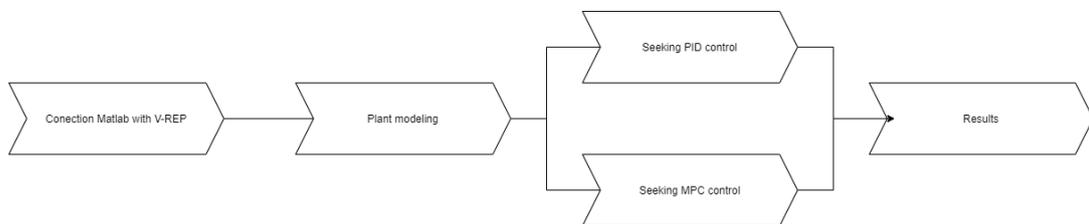


Figura 3: Outline of the procedure to be followed in the project

4. Results

The plants obtained for the behavior of the robot in free navigation are quite simple, with a single-pole and a gain. Therefore, the design of the PID control seems quite simple, since the possibilities of P, PI, and PID control should only be studied.

The wall tracking plant is much more complicated, and due to its shape with positive zero, it was decided to proceed to the design of cascade control, working with two different controls: P control in the internal loop and PD control in the external loop and vice versa. Figure 4 shows the result of the cascade control for PD on the inside and P on the outside.

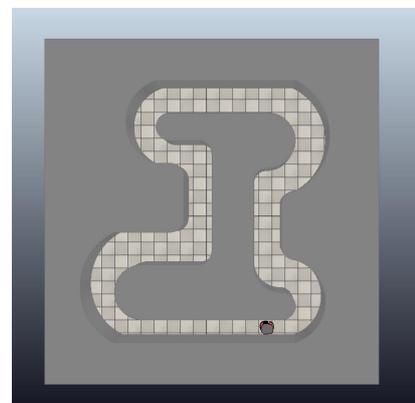
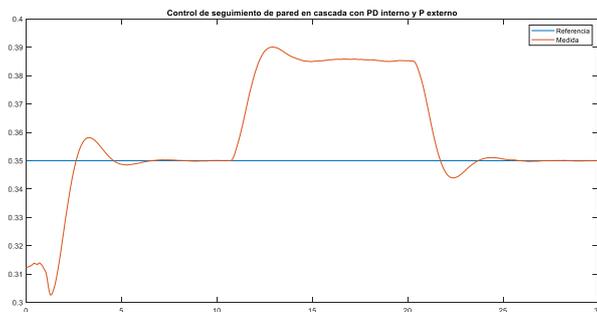


Figura 4: Optimal result, cascade control for wall tracking plant and simulation environment

5. Conclusions

When comparing the different PID controls that have been designed, it was decided to exclusively use the cascade control with PD control on the internal loop and P on the external, as it is the most efficient. When trying to work with control for the forward speed plant and for the raw speed plant at the same time as the control of the wall-tracking plant, it is observed that they are of different speeds and that they do not follow each other well.

Since no results have been obtained in the predictive control, it is taken into account as a possible future work, in which different external factors not measurable are taken into account. This project would be feasible in an academic context.

6. References

- [1] del Val Román, J. L. (2016). *Industria 4.0: la transformación digital de la industria*. In Valencia: Conferencia de Directores y Decanos de Ingeniería Informática, Informes CODDII. <http://coddii.org/wp-content/uploads/2016/10/Informe-CODDII-Industria-4.0.pdf> (Last access: 05/06/2020)
- [2] D'Andrea, R. (2012). Guest editorial: *A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead*. *IEEE Transactions on Automation Science and Engineering*, 9(4), 638-639. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6295687> (Last access: 19/08/2020)

Índice de la memoria

Capítulo 1. Introducción	7
Capítulo 2. Descripción de las tecnologías	13
Capítulo 3. Estado de la cuestión	17
Capítulo 4. Definición del trabajo.....	21
4.1 Justificación.....	21
4.2 Objetivos	21
4.3 Metodología.....	21
4.3.1 Vinculación del programa de simulación del robot y Matlab.....	21
4.3.2 Modelo de la planta.....	23
4.3.3 PID	24
4.3.4 Predictivo	24
4.4 Planificación.....	25
Capítulo 5. Modelo de la planta	27
5.1 Análisis del Sistema	27
5.2 Planta para el Modelo de Navegación Libre	28
5.2.1 Estudio de planta para velocidad de avance.....	29
5.2.2 Estudio de planta para velocidad de giro	31
5.3 Planta para el modelo de navegación de seguimiento de pared	33
Capítulo 6. Control PID	37
6.1 Control PID como concepto	37
6.2 Forma del PID utilizado	38
6.3 Diseño del PID	42
6.3.1 Control PID para planta de velocidad de avance	42
6.3.2 Control PID para planta de velocidad de giro.....	47
6.3.3 Control PID para planta de modelo de seguimiento de pared.....	50
Capítulo 7. Control predictivo	57

7.1	Control predictivo como concepto	57
7.2	Funcionamiento y diseño del control predictivo utilizado	58
7.3	Diseño del control predictivo	60
Capítulo 8. Análisis de Resultados.....		63
8.1	Resultados obtenidos con el control PID.....	63
Capítulo 9. Conclusiones y trabajos futuros.....		69
Capítulo 10. Bibliografía.....		71
ANEXO I: FreeCad con V-REP		75

Índice de figuras

Figura 1: Esquema del procedimiento a seguir en el proyecto.....	10
Figura 2: Control en cascada para planta de seguimiento a la pared y entorno de simulación	11
Figura 3: Outline of the procedure to be followed in the project	14
Figura 4: Optimal result, cascade control for wall tracking plant and simulation environment	14
Figura 5. Historia de la industria según las diferentes revoluciones industriales [1]	8
Figura 6: Esquema explicativo de conexión de las tres herramientas	15
Figura 7: Escena V-REP estudio control para planta de seguimiento a pared.	16
Figura 8: Concepto físico del robot Pioneer P3DX [12]	22
Figura 9: Esquema explicativo funcionamiento de los scripts de conexión Matlab con V-REP	23
Figura 10: Gráfico comparativo de ensayos de velocidad de avance 1 y 2.....	30
Figura 11: Gráfico comparativo de ensayos de velocidad de giro 1 y 2	32
Figura 12: Explicación de las distancias y ángulos medidos según los sensores utilizados [13]	33
Figura 13: Modelo linealizado planta de seguimiento a pared.....	35
Figura 14: Diagramas de Nyquist y Black [15].....	40
Figura 15: Control PID no interactivo con 2 grados de libertad [13].....	41
Figura 16: Resultado del control P para la planta de velocidad de avance	43
Figura 17: Resultado del control PI para la planta de velocidad de avance	45
Figura 18: Resultado del control PID para la planta de velocidad de avance	46
Figura 19: Resultado del control P para la planta de velocidad de giro	47
Figura 20: Resultado del control PI para la planta de velocidad de giro.....	48
Figura 21: Resultado del control PID para la planta de velocidad de giro.....	50
Figura 22: Control de un único lazo (a) y control en cascada (b)[13].....	51

Figura 23: Diseño final por frecuencia del control P externo del control cascada.....	54
Figura 24: Diseño final por frecuencia del control PD externo del control cascada.	56
Figura 25: Diagrama de bloques de un control predictivo	58
Figura 26: Estructura del funcionamiento del control MPC 0.....	60
Figura 27: Definición de las variables de entrada y de salida	61
Figura 28: Gráfica de entrada y salida del control predictivo	62
Figura 29: Representación de los controles clásicos para plantas de velocidad de avance y giro.....	64
Figura 30: Gráfica de la distancia del robot con la pared con el control en cascada en simulación.....	65
Figura 31: Entorno de Vrep para la simulación final del control	67

Índice de tablas

Tabla 1: Parámetros control de seguimiento a pared óptimo	10
Tabla 2: Planificación del proyecto	25
Tabla 3. Parámetros planta para velocidad de avance	30
Tabla 4. Parámetros planta para velocidad de giro.....	32
Tabla 5: Recurso explicativo del comportamiento del PID según sus parámetros	41
Tabla 6: Parámetros del control P para la planta de velocidad de avance.....	44
Tabla 7: Parámetros del control PI para la planta de velocidad de avance	45
Tabla 8: Parámetros del control PID para la planta de velocidad de avance.....	46
Tabla 9: Parámetros del control P para la planta de velocidad de giro	48
Tabla 10: Parámetros del control PI para la planta de velocidad de giro	49
Tabla 11: Parámetros del control PID para la planta de velocidad de giro	50
Tabla 12: Parámetros control PD interno del control en cascada.....	53
Tabla 13: Parámetros control P interno del control en cascada.....	56
Tabla 14: Valores medios según el control utilizado para planta de velocidad de avance y giro.....	64
Tabla 15: Parámetros del control final	66

Capítulo 1. INTRODUCCIÓN

La robótica es un área de investigación relativamente moderna, pues parte de otras ramas de investigación y desarrollo mucho anteriores. Lo que conocemos hoy en día como robótica se ha ido desarrollando a lo largo de la historia, comenzando con la primera Revolución Industrial, que tuvo lugar durante los siglos XVIII y XIX. Este período se caracterizó por la mecanización de los procesos de producción, lo que conllevó transportar a la economía agraria y artesanal a un segundo plano. De este modo la industria se posicionó en primer puesto a nivel económico [1].

Después de esta primera revolución y a medida que se producía el avance de estas tecnologías se pasó por otras dos etapas a lo largo del siglo XX. La primera de ellas se caracterizaba por la producción en serie de productos destinados al gran consumo. La segunda sucedió a finales del siglo con la aparición de la electrónica y la informática en los procesos industriales [1].

Una vez establecidas y desarrolladas éstas últimas tecnologías y entrando ya en el siglo XXI, con el avance de Internet, tanto la economía como la sociedad se han visto muy afectadas por este último y moderno desarrollo. Fue precisamente Internet lo que llevó a los profesionales a declarar un nuevo periodo tecnológico: la Cuarta Revolución Industrial o Industria 4.0, que fue el término que se acuñó de manera oficial en la Feria de Hannover de 2011, en Alemania. Con el objetivo de resumir de manera visual cómo ha ido evolucionando la industria hasta entender lo que tenemos ahora se presenta la Figura 5 [1]:

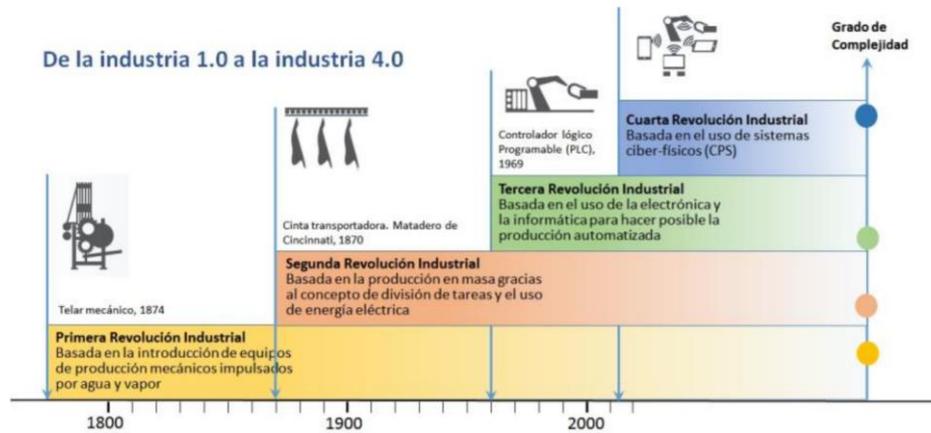


Figura 5. Historia de la industria según las diferentes revoluciones industriales [1]

Y así es como se ha llegado a la etapa actual, etapa de conexión y de industria “inteligente”. La industria 4.0 hace referencia a una nueva forma de organización que ha sido posible gracias a lo que se conoce como tecnologías de la información, es decir, el uso de equipos y ordenadores con el objetivo de guardar, manipular, recuperar y transmitir datos. Este fresco término de industria 4.0 recoge una serie de términos que han aparecido dentro de este contexto tecnológico. Entre estos destacan los dos siguientes:

- IoT/Internet of Things o “Internet de las cosas”: este concepto aparece en 1982 cuando una máquina de refrescos que modificada para que indicase el almacenamiento actual y la temperatura de los refrescos. El objetivo que persigue el IoT es permitir el intercambio de información entre elementos conectados de manera invisible y que deciden tomar decisiones y transmitir información en función de lo que sea capturado por los sensores [2][1].
- “Fábrica Inteligente”: este término hace referencia a las industrias que se mueven según un modelo de IoT, lo que quiere decir que la información de sensores, de robots e información obtenida a través de la informática convergen para dar lugar a espacios ciber físicos. Para dar lugar a estos sistemas, se necesita la ayuda de aplicaciones como el Big Data y M2M, que serán explicadas más adelante.

Un buen resumen que agrupa estos tres términos sería mencionar que todos ellos describen unos modelos industriales en los que la principal característica es la comunicación que se lleva a cabo dentro de la industria. Todos los sistemas suelen estar interconectados, en su gran mayoría, de manera inalámbrica y gracias a Internet.

En este nuevo modelo de industria aparecen términos que describen problemas y soluciones que hacen referencia a situaciones persistentes en el día a día de una industria conectada. Entre todas ellas son destacables:

- Comunicaciones móviles: derivado de que no todos los elementos que trabajan son estáticos, se crea la necesidad de que a la hora de coordinarse (M2M) puedan recibir información unos de otros sin la necesidad de estar físicamente conectados. Se habla de comunicación o internet móviles y es la base del IoT [1].
- Big data: como su nombre indica, define la cantidad enorme de datos que deben ser tratados. Pero este término ha ido un poco más allá, y ya no hace referencia sólo a la descripción de estos datos sino a las herramientas que son utilizadas con el objetivo de organizarlos y descubrir posibles comportamientos futuros gracias a observar estos datos pasados [3].
- Cloud computing: desde la aparición de internet se ha evolucionado desde aplicaciones que debían ser descargadas para su posterior uso hasta el día de hoy, que aparecen aplicaciones que sin necesidad de actualización están todo el día actualizadas. Esto se debe al *cloud computing*, que ha potenciado y ayudado tanto a empresario como consumidor. Suele ser más sencillo, especialmente para el consumidor, pues no requiere saber de su infraestructura para trabajar con la nube y en el caso de que aparezca algún problema, este servicio es autorreparable. Además, la aparición del *cloud computing* ha sido muy beneficioso especialmente a medida que han ido surgiendo aparatos de menor capacidad que son capaces de trabajar con *cloud computing* (como puede ser un *smartphone* y una *tablet*) [4].
- Comunicación M2M: sería la tecnología base si se hablase de alguna dentro del IoT. Como se ha explicado anteriormente, en el IoT es esencial la comunicación entre

elementos especialmente sin cableado. A esto hace referencia la comunicación M2M (*machine to machine*), en la que las tecnologías apropiadas recopilan información de la máquina o dispositivo deseado y ésta es enviada a otra máquina o dispositivo con el objetivo de que actúe de una manera determinada por los datos recibidos.

- Realidad aumentada: actualmente se encuentran todavía en proceso de desarrollo, pero esta herramienta será capaz de ayudar a trabajadores y hacer su trabajo más rápido y visual. Consiste en la creación de imágenes o sonidos a través de dispositivos (por ejemplo, gafas) mediante los cuales y dependiendo del espacio en el que se esté se pueden enviar ordenes o interactuar con el espacio [1].
- Robótica avanzada y colaborativa: todavía en desarrollo, los robots que se pretenden ir integrando en la industria deben ser cada vez más autónomos y colaborativos. A nivel electrónico no es tan visible la evolución realizada en los últimos años, y puesto que a nivel sensorial y actuadores estos robots suelen estar bastante bien equipados, se pretende participar más en la mejora de las inteligencias de éstos mismos [1].

Como bien se ha destacado en el modelo de Industria 4.0 aparecen diferentes herramientas para solucionar los problemas de una manera más virtual de lo que se haría siglos atrás, pero este proyecto se centrará en la parte de los robots.

Los robots que participan dentro de la Industria 4.0 son principalmente robots autónomos. Tanto móviles como fijos, estos robots cumplen una parte importante dentro del proceso industrial actual. En el nuevo modelo las comunicaciones móviles son una parte muy importante, lo cual viene relacionado con la comunicación M2M (*machine to machine*). Los aprendizajes por refuerzo y predictivos son esenciales, pues debido a la cantidad de datos que se generan en las industrias, se pueden desarrollar modelos predictivos en vez de preventivos, que pueden aplicarse en todas las áreas. Esto quiere decir que en vez de que un robot sea equipado con reacciones ante un error, el modelo va a ser trabajado de manera que detectará el error antes de que éste aparezca, por lo que la preparación será mayor [5].

Una vez contextualizado el proyecto, se debe explicar la motivación que ha impulsado este proyecto. Una de las principales razones por las que se ha decidido realizar el proyecto ha sido la utilización de los desarrollos conseguidos con el trabajo realizado con el objetivo de mejorar la docencia de las asignaturas de robótica móvil.

Otro de los motivos por los que se decide hacer este proyecto tiene que ver con el estudio en comparativa de controles más clásicos con controles más modernos. Éstos primeros llevan más años en uso y más estudios realizados, pues llevan más tiempo implantados en la industria y son los controles más utilizados. Los controles más modernos son los más relacionados con la industria 4.0, aquellos que se basan en algoritmos de aprendizaje por refuerzo.

Por último, este proyecto tiene como impulso el hecho de que se realiza exclusivamente en ordenador, y al ser realizado exclusivamente con simulaciones no tiene gran peso económico ni ambiental, pues los programas utilizados son versiones académicas pensadas precisamente para este tipo de trabajos.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

Las tecnologías utilizadas en este proyecto no requieren de conocimiento previo, aunque puede ayudar. Teniendo en cuenta los objetivos que se plantean y buscando la manera más sencilla de cumplirlos, serán necesarias las siguientes herramientas:

- Matlab. En proyectos del estilo, es una herramienta sencilla que se utiliza para conseguir una planta del modelo y para escoger los mejores algoritmos de los controles. Además, son necesarias las toolboxes ‘Model Predictive Control Toolbox’ y ‘Control System Toolbox’, para el diseño del control predictivo.
- V-REP/CoppeliaSim. Se trata de una herramienta de simulación, en este caso utilizada para trabajar con un robot móvil, que ayuda a visualizar el funcionamiento de éste y permite tomar datos del robot para encontrar un control óptimo. Además, es muy útil a la hora de realizar la planta del modelo. Respecto al software de *CoppeliaSim* [6], es un programa previamente conocido como V-REP, con muchas características y aplicaciones diferentes. Aunque el software completo es capaz de hacer muchas cosas, el utilizado para este proyecto es la versión de estudiantes, y será suficiente para realizar el estudio que se quiere llevar a cabo.
- FreeCad. Es un programa utilizado para el diseño de piezas en 3D, en esta ocasión en formato STL. Se utiliza para el diseño de los entornos que se utilizan para poner a prueba los controles.

A pesar de que Matlab es una herramienta conocida, los comandos API de conexión con V-REP no lo son tanto. Por esto, las funciones utilizadas en los códigos de vinculación quedan explicadas a continuación. La disposición de ellas en el código se explica de manera esquemática en la Figura 6:

- *simxFinish (-1)*: línea de código utilizada para terminar la simulación. Aparece al principio del código, con el objetivo de cerrar una posible simulación que no hubiese terminado y al final de este, con el objetivo de terminar ésta misma [6].

- *simxStart ('127.0.0.1', 19997, True, True, 2000, 5)*: comando que permite conectar con V-REP y empezar la simulación. La información que transmitirle al comando debe ser, por orden, dirección de conexión (*string*), número de puerto de conexión (*number*), esperar hasta conectado (*boolean*), no reconectar una vez desconectado (*boolean*), tiempo máximo de conexión para el primer intento (*number*) y ciclo logrado o número de veces que se envían los paquetes de datos y se reciben (*number*). Esta función devuelve lo que se conoce como *client_id*, que es el identificador de la simulación, necesario para casi todas las demás funciones [6].
- *simxSynchronous (client_id, True)*: habilita o deshabilita (*True* o *False*) el modo síncrono del API al que está conectado, en función del *boolean*. Aparece al principio del código una vez se ha obtenido un *client_id* [6].
- *simxStartSimulation (client_id, simx_opmode_blocking)*: solicita el comienzo de la simulación. Se utiliza en *blocking mode* porque se quiere esperar a la respuesta de verdad respecto al comando mandado [6].
- *simxSynchronousTrigger (client_id)*: indica al programa cuando realizar el próximo paso de la simulación. Es necesario utilizar la función *simxSynchronous* antes de utilizar esta otra. Aparece dentro del bucle cada vez que se quiere mandar información desde Matlab a V-REP [6].
- *simxGetPingTime (client_id)*: recupera el tiempo para que el comando sea enviado, ejecutado y mandado de vuelta. Aparece dentro del bucle cada vez que se quiere mandar información desde Matlab a V-REP [6].
- *simxStopSimulation (client_id, vrep.simx_opmode_blocking)*: solicita una interrupción de la simulación. El modo recomendado para este comando es *blocking* [6].
- *simxPauseCommunication(client_id, True)*: interrumpe la comunicación momentáneamente para no enviar datos. Suele ser utilizado para enviar varios datos al simulador y que lleguen a la vez [6].

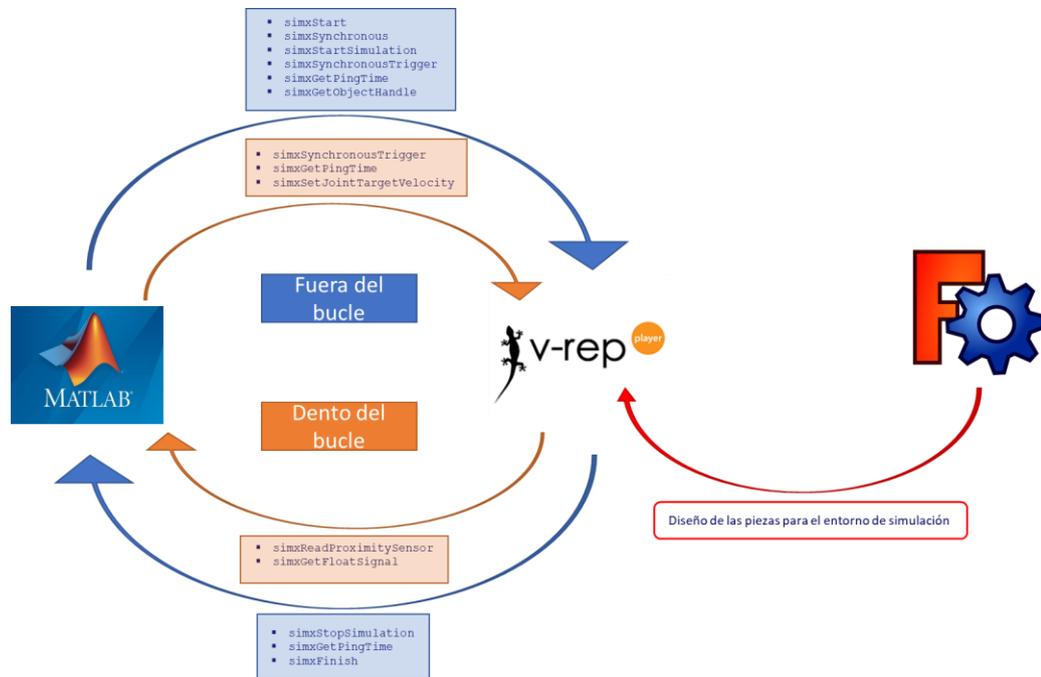


Figura 6: Esquema explicativo de conexión de las tres herramientas

Otras funciones que también serán necesarias, no tanto para simplemente conectar con V-REP, sino para enviar y obtener información del robot, serían:

- *simxSetJointTargetVelocity (client_id, motors('left'), wl, simx_opmode_oneshot)*: es el comando que se encarga de establecer una velocidad, siempre pasando el dato de manera angular, en el robot. En este caso *motors('left')* está haciendo referencia al mando del motor izquierdo [6].
- *simxReadProximitySensor (client_id, sensors(i), simx_opmode_buffer)*: lee el valor que capturan los sensores de proximidad (en el caso del robot escogido lee 16 sensores cada vez que se llama a esta función). El vector *sensors(i)* hace referencia a cada uno de los *handles* de los sensores de proximidad. Éstos han sido clasificados de 1 a 16 [6].
- *simxGetFloatSignal (client_id, char(string(encoder) + 'Encoder'), simx_opmode_buffer)*: toma la información de la señal que se le mande, en este caso del *encoder* izquierdo. Devuelve el valor de la velocidad de giro de la rueda izquierda. Aparece también como comando para declarar que se quiera usar esa

señal. En este caso, la manera de iniciar los *encoders* sería: *simxGetFloatSignal(client id, 'leftEncoder', simx_opmode_streaming)* [6].

- *simxGetObjectHandle(client id, char('Pioneer_p3dx ' + string(motor) + 'Motor'), simx_opmode_blocking)*: Obtiene el *handle* del objeto especificado en la función. El nombre declarado del objeto tiene que ser el mismo que el que aparece en el programa de V-REP. Esta función será usada tanto para los motores como los sensores de proximidad [6].

El programa tiene diferentes opciones respecto al espacio de simulación. Lo primero de todo a la hora de crear el espacio sería añadir sobre éste el robot que se quiera simular, en este caso será el Pioneer P3DX. Sobre este mismo se le añaden todas las funciones extra que queramos que tenga nuestro robot (como pueden ser por ejemplo los sensores de proximidad) y por último se caracteriza toda la escena. El resultado del entorno creado para la comprobación del control en la planta de seguimiento a pared es el expuesto en la Figura 7.

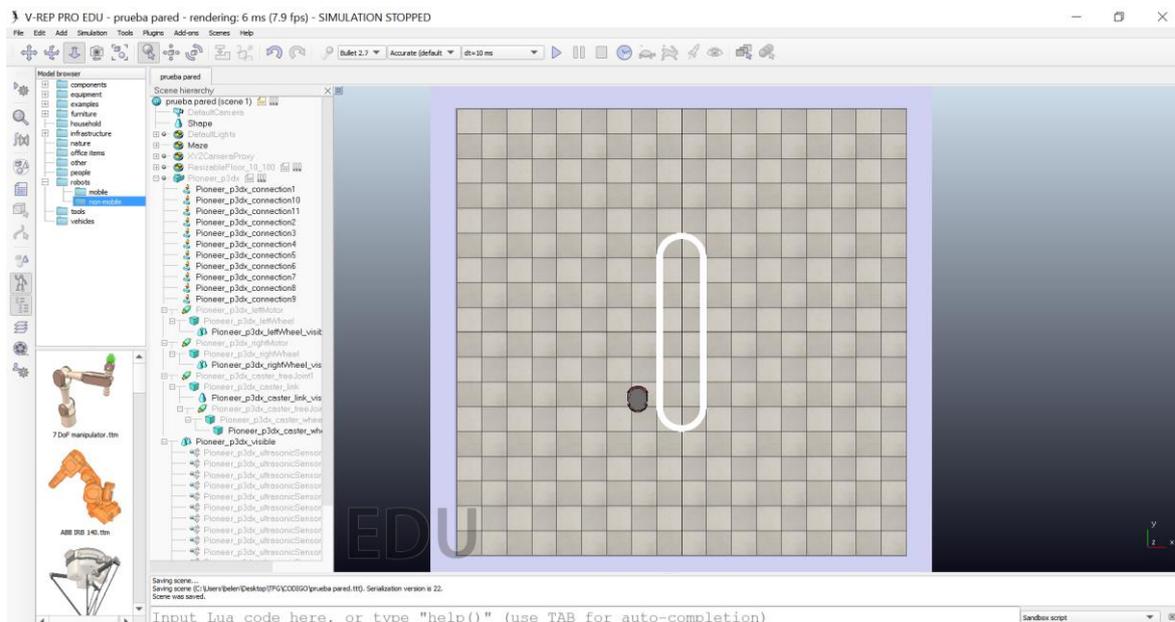


Figura 7: Escena V-REP estudio control para planta de seguimiento a pared.

Capítulo 3. ESTADO DE LA CUESTIÓN

Los robots como se conocen hoy en día están a disposición del ser humano y facilitan multitud de tareas. Estos mismo son capaces de trabajar en espacios abiertos y en espacios cerrados y son diferentes según la tarea de la que se vayan a encargar. Una buena forma de clasificarlos sería, entre otras, robot de exteriores y robots de interiores.

Entre los robots de exteriores, suelen destacar aquellos que son móviles, y destacan porque tienen funciones muy dispares, como podrían ser una máquina cortacésped que se guía por posicionamiento GPS, un robot militar que es un tanque no tripulado e incluso un vehículo con brazo móvil que sea capaz de recoger basura. Al estar hablando de robot exterior, lo más común es que los robots sean móviles, y se dirijan por posicionamiento GPS.

Centrando el tema en los robots de interiores, se expanden las posibilidades, pues aparecen en áreas diferentes: a nivel industrial, a nivel comercial e incluso a nivel doméstico. En esta ocasión, suelen trabajar más de la mano del hombre, es decir, ambos trabajos se suelen complementar.

Actualmente, los robots fijos de interiores que se conocen son mayoritariamente los que trabajan en la industria. Son realmente necesarios por sus grandes capacidades físicas a la hora de levantar pesos excesivos para el ser humano y por sus capacidades en precisión. El ser humano comparte con ellos el entorno, pero a la hora de realizar tareas suelen ir por caminos diferentes pues mientras el robot realiza el trabajo que da lugar a un producto, la persona se dedica a vigilar que la máquina y sus productos estén en un buen estado.

Aunque los anteriores sean muy necesarios en el proceso industrial, los móviles están cada vez más al orden del día. Igual que los anteriores tenían características de fuerza y precisión, éstos de ahora tienen que seguir manteniendo la precisión y además se le añade la inteligencia y el sentido de la orientación. En este sentido aparece una parte de complicación,

pues igual que los que eran de exteriores podían hacer eso del GPS, estos necesitan otro tipo de posicionamiento que posibilite la orientación dentro del circuito cerrado.

Entre las distintas posibilidades de orientación de un robot en el espacio, al hablar de un interior, aparecen la creación de mapas de interiores que permiten que el robot se oriente dentro del espacio creado. Para esto surgen diferentes algoritmos, entre los que destaca SLAM (*simultaneous localization and mapping*). Si en el principio de los tiempos se diseñaba el mapa y más adelante se programaba el robot para que éste fuese capaz de localizarse en el entorno, gracias a esta técnica los robots son capaces de crear y orientarse en el espacio de manera simultánea.

A la hora de tomar la decisión de desplazarse, los robots necesitan características del entorno para tener claro el rumbo que decidan seguir. Para ello, se ayudan de sensores que les permitan captar la información de los que les rodea. Entre los sensores más comunes que suelen tener este tipo de robots destacan:

- Sensor láser: son utilizados para obtener información respecto a las distancias con los posibles obstáculos. Son muy precisos, pero cubren un rango muy pequeño de distancia. Son muy útiles cuando el desplazamiento que debe realizar el robot es muy pequeño (inferior a 1 metro de distancia del obstáculo) y debe ser muy preciso. También es utilizado como complemento para trabajar con otros sensores, como puede ser a la hora de utilizar una cámara para ayudar a mantener mejor la referencia con el objeto.[7]
- Sensor ultrasonido: son utilizados en multitud de proyectos por la relación calidad precio. La información obtenida es bastante exacta. En ocasiones, la información que se aporta por el fabricante respecto a ellos es escasa, por lo que son necesarias pruebas previas para obtener los alcances. Son muy útiles cuando lo principal es la referencia de distancia con los obstáculos. [8]
- Cámara: son las que más estudios están recibiendo en los últimos años. Tienen muchas posibilidades, pues por medio de algoritmos de reconocimiento, se pueden detectar objetos e indicar a que distancia se encuentran. Gracias a *VisualSlam*, que

es la técnica de creación de un mapa y orientación de manera simultánea gracias a la obtención de las imágenes del entorno por medio de una cámara, se consiguen resultados asombrosos.

- Sensor de presión o fuerza: más que esquivar los obstáculos, con estos sensores se consigue descubrirlos al colisionar. Este es el caso de *Roomba*, la aspiradora autónoma de *iRobot*, que es capaz de desplazarse por un espacio cerrado sin quedarse parada al colisionar con un objeto.

Si se toma como referencia los robots más visibles en la industria por sus grandes capacidades y avances tecnológicos, se debe destacar el robot autónomo adquirido por la empresa de *Amazon*. En el 2003, la empresa decidió crear su propio departamento de robótica conocido como *Amazon Robotics*, que se encarga del desarrollo de productos robóticos, en su mayoría utilizados para mejorar y desarrollar tecnológicamente sus almacenes.

Entre todas las implementaciones que han desarrollado, cabe destacar el robot *Kiva*. Este robot de 40 cm de altura es capaz de desplazar pesos de hasta 1 tonelada. La implementación de este robot a los almacenes facilita el orden de las estanterías de almacenamiento, a la vez de que reduce el espacio ocupado por cada *pod*/estantería. El objetivo de este dispositivo es desplazar las estanterías hasta cada trabajador, en lugar de que cada trabajador se desplace hasta ellas [9].

Es interesante todo lo que consigue hacer el robot, pero aún lo es más el cómo lo hace. Por una parte, consta de dos cámaras, pero en esta ocasión la función de éstas es solo la lectura de códigos que hace que decida coger una estantería o no. Es el sensor de infrarrojos el que participa en la detección de un objeto desconocido u obstáculo, que activaría el sistema de prevención de colisiones parando el dispositivo [4] [10].

Pero no sólo es la empresa puntera de estados unidos la que hace uso de robots móviles para acelerar el ritmo de trabajo en su empresa y optimizar el tiempo y espacio, pues en la fábrica de SEAT situada en Martorell, hace bien poco que tuvieron la idea de adquirir robots

autónomos con el objetivo de poder desplazar las piezas más grandes por fuera del taller sin la necesidad de que alguien tenga que estar pendiente de ellas.

El sistema escogido emplea técnicas de *SLAM*, con conexión 4G. Es capaz de mantenerse dentro de una carretera y detectar un obstáculo (bien puede ser una persona, bien otro coche) para frenar y no colisionar con él. [11]

Observando todo lo que se desarrolla a nivel tecnológico y todo lo que no se conoce, tendría mucho sentido que se realizasen gran cantidad de estudios sobre el tema. Una vez presentados proyectos de robótica interior y habiendo investigado respecto a los distintos proyectos que pusiesen relación entre los robots móviles y sus respectivos sensores controles utilizados, la información obtenida ha sido bastante escasa. Casi toda la información que se ha recopilado y expuesto en estudios y revistas tecnológicas es de cómo estos robots obtienen información del entorno (sensores), pero no de cómo trabajan con ella (controles).

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Mediante la investigación de artículos y estudios que se han encontrado a la hora de formarse sobre los robots de interiores y su funcionamiento, la información obtenida ha sido escasa. Especialmente a la hora de hablar de los controles que se utilizan habitualmente. Toda la información recopilada explica superficialmente que se utilizan técnicas de *machine learning* o incluso de aprendizaje por refuerzo para crear los controles de estos robots.

Es por todo esto que se ha decidido realizar un proyecto en el que se ponga en comparación dos tipos de controles, en los que ninguno utilice técnicas de *machine learning*, y se compare un control clásico como puede ser el PID, con uno más complejo como es el predictivo.

4.2 OBJETIVOS

El objetivo de este proyecto es ajustar algoritmos de control para conseguir que un robot móvil con cinemática diferencial sea capaz de recorrer un entorno siguiendo la pared, sin colisionar, alcanzando una velocidad de 0.5 m/s.

4.3 METODOLOGÍA

El objetivo del proyecto es conseguir que los algoritmos de control creados permitan que el robot se desplace evitando obstáculos a una velocidad constante de 0.5 m/s. Para llegar a ese objetivo final, es necesaria cierta estructura y orden a la hora de trabajar.

4.3.1 VINCULACIÓN DEL PROGRAMA DE SIMULACIÓN DEL ROBOT Y MATLAB

Al trabajar con un programa no conocido previamente, se requiere tiempo de formación para entender las funcionalidades de éste. Además, es necesaria la creación de un código en

Matlab que conecte un programa con otro, pues desde Matlab se enviarán las órdenes al programa para que éste reciba y actúe en consecuencia.

Es importante mencionar las características del robot con el que se trabaja. El robot escogido es el PioneerP3DX. Consta de dos ruedas móviles de 19.5 cm de diámetro separadas por un eje de 33 cm y una rueda loca que no es manejable. Además, para este experimento se le han añadido al vehículo 16 sensores ultrasónicos, de los cuáles serán utilizados exclusivamente dos de ellos (sensores 8 y 9). Estos dos sensores están separados entre ellos por una distancia de 216.7 mm y la señal que recogen es paralela entre ellos y perpendicular al cuerpo del robot. Así se muestra en la Figura 8.

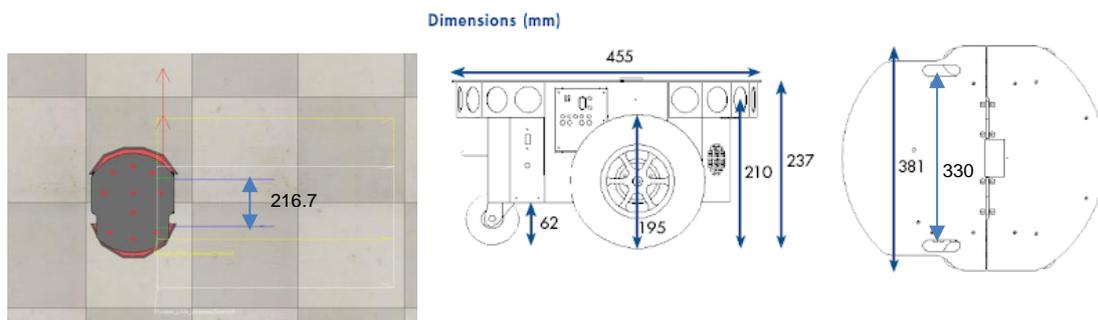


Figura 8: Concepto físico del robot Pioneer P3DX [12]

El código está compuesto de tres scripts diferentes, tal y como se explica de manera esquemática en la Figura 9. Por un lado, aparece una función abstracta conocida como Robot, que llama a otra conocida como RobotP3DX que contiene las características del robot que se está utilizando, la manera de recoger la información y la manera de desplazarse. Por último, es necesario un script conocido como *main*, que es el que enciende y apaga el programa de simulación, y que crea un objeto conocido como robot, que es creado gracias a la función de RobotP3DX, y gracias a esta función se consigue realizar todas las acciones que se necesitan para desplazar a éste y recopilar datos.

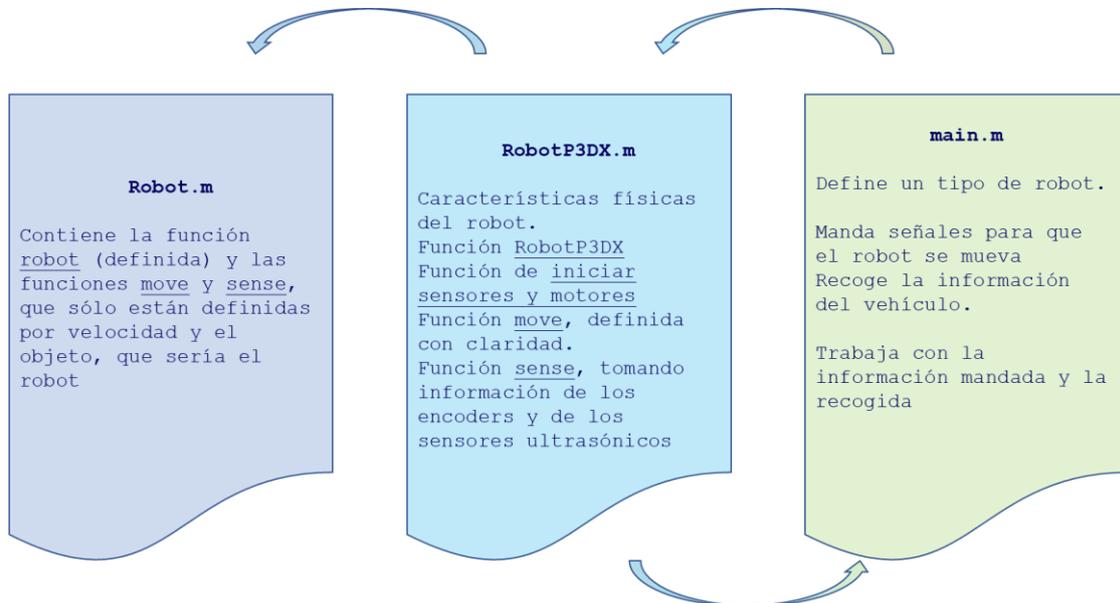


Figura 9: Esquema explicativo funcionamiento de los scripts de conexión Matlab con V-REP

4.3.2 MODELO DE LA PLANTA

Para más adelante poder trabajar con los controles, se necesita obtener una función de transferencia que explique el modelo comportamiento ante las diferentes posibilidades. Serán necesarias tres plantas diferentes:

- Planta de velocidad de avance
- Planta de velocidad de giro
- Planta de seguimiento a pared

Las dos primeras se realizan de manera similar. Primero se recogen datos del entorno (velocidad de avance y de giro de referencia y de salida) y después, mediante el método de mínimos cuadrados se obtiene una función de transferencia. En todo este proceso se está trabajando en tiempo discreto, pues el tiempo de muestreo es relativamente grande. Cuando se compila el *script* de los mínimos cuadrados se obtiene directamente los parámetros de la función de transferencia, pero es importante conocer previamente la forma de ésta. En este proyecto, la forma de la planta consta exclusivamente de una constante de tiempo y de una ganancia.

La otra planta se obtiene mediante la planta de velocidad de giro y mediante la información física del posicionamiento de ellos sensores y la velocidad media a la que se quiere desplazar el vehículo.

4.3.3 PID

Una vez obtenidas las plantas, se procede al análisis del control de cada una de ellas. Para ello se echa mano del programa de GUI_control_digital.m creado por Juan Luis Zamora, profesor de ICAI, mediante el cual se pueden realizar diferentes ajustes observando los efectos en la respuesta y en los polos.

En las dos primeras plantas se realizará un estudio para observar cuál de todos los controles funciona mejor, incluyendo en estos el proporcional y el integral, además del PID.

La planta de seguimiento a pared tiene un estudio más exhaustivo, pues por su configuración con un cero positivo y con dos integraciones, es necesario un control en cascada, para lo que es necesario configurar primero por señales discretas y más adelante utilizar este mismo programa utilizado para el establecimiento de los otros controles de las plantas.

Por último, se realiza una prueba en circuito para analizar el comportamiento de los controles en un entorno más realista.

4.3.4 PREDICTIVO

Se realiza un breve estudio de cómo debería procederse para diseñar el control predictivo y se estudian las variables a tener en cuenta. Se toma como ayuda el proyecto de Emil Hernández-Arroyo, Jorge Luis Díaz-Rodríguez y Omar Pinzón-Ardila, Estudio del comportamiento de un control MPC comparado con un control PID en una planta de Temperatura [13].

4.4 PLANIFICACIÓN

A continuación, en la Tabla 2 se presenta la planificación llevada a cabo para la realización del proyecto.

REPARTO DE TAREAS	04/05	11/05	18/05	25/05	01/06	08/06	15/06	22/06	29/06	06/07	13/07	20/07	27/07	03/08	10/08	17/08	24/08
Anexos iniciales																	
Anexo A																	
Anexo B																	
Proyecto																	
Código conexión Matlab-Vrep																	
Planta y controles																	
Datos desde Vrep a Matlab																	
Planta y modelado																	
PID teórico																	
PID código																	
Predictivo teórico																	
Predictivo código																	
Memoria del proyecto																	

Tabla 2: Planificación del proyecto

Capítulo 5. MODELO DE LA PLANTA

En este capítulo se presenta el modelo matemático. Para ello, se realizan diferentes ensayos y se recogen los datos medidos por los sensores con el objetivo de obtener una función de transferencia que sirva de modelo matemático para trabajar más tarde con los controles del vehículo.

5.1 ANÁLISIS DEL SISTEMA

El robot escogido para ensayar es el Pioneer P3DX, dispositivo muy práctico para desarrollar ensayos y pruebas de todo tipo, pues consta de varias herramientas y opciones de sensores que se le pueden añadir según sea necesario. Puesto que el objetivo del proyecto es el desarrollo de algoritmos de control de un robot móvil que se desplace según seguimiento a pared, es necesario el estudio de 3 plantas diferentes, clasificadas en dos grupos:

Plantas de navegación libre:

- Planta de velocidad de avance: se realiza su estudio porque se quiere mantener el robot a una velocidad constante durante el tiempo que éste se esté desplazando.
- Planta de velocidad de giro: se requiere el estudio del comportamiento del sistema cuando éste quiere efectuar algún giro. Es estudiado porque se requiere que no realice ningún giro a la hora de realizar el estudio. Todo giro se realiza en función de la distancia a la pared.

Planta de seguimiento de pared:

- Planta de seguimiento a la pared: se estudia con el objetivo de obtener un comportamiento acertado de la relación entre la velocidad angular del dispositivo y la distancia a la pared.

Para realizar los respectivos ensayos y obtener una función de transferencia acertada que ayude a entender el comportamiento en las diferentes condiciones impuestas, se ha utilizado la ayuda de Matlab, en concreto de unos programas realizados previamente por Juan Luis Zamora, profesor de ICAI con gran experiencia en la regulación automática. Estos códigos han sido obtenidos del laboratorio de Control avanzado de ICAI.

Además de los códigos para trabajar con los datos obtenidos, cabe destacar que el entorno de simulación utilizado en V-REP es un entorno abierto, sin obstáculos que se interpongan en el movimiento del vehículo.

5.2 PLANTA PARA EL MODELO DE NAVEGACIÓN LIBRE

Cuando se habla de navegación libre, se considera que ningún factor del entorno está afectando a las decisiones tomadas por el robot. Por tanto, en el estudio de estos comportamientos las variables medidas son aquellas que no dependen del entorno, es decir, velocidad de avance y velocidad de giro del robot. En este caso, las variables son obtenidas gracias a los *encoders* que se encuentran en las ruedas del robot que está siendo simulado.

En la navegación libre que se trabaja en este proyecto, las plantas que se obtienen son sencillas, de un solo polo y una ganancia, porque el sistema consta de una ganancia y de una constante de tiempo que define el modelo del vehículo. En otros proyectos parecidos podría aplicarse un retraso ρ , con la intención de indicar la diferencia entre el momento de mandar la acción y la realización de ésta. Al no estar trabajando con un robot físico sino con una simulación del robot, no es necesario añadirlo, pues en la simulación todo es instantáneo. En el caso de considerarse necesario, podría simularse también el retardo. Por tanto, la ecuación resultante de la función de transferencia de cada planta para el modelo de navegación libre sería la siguiente:

$$sys = \frac{K}{T * s + 1}$$

Ecuación 1

5.2.1 ESTUDIO DE PLANTA PARA VELOCIDAD DE AVANCE

La planta para velocidad de avance es la primera las plantas de navegación libre que se modela. Para lograr un modelo acertado se deben seguir unos pasos determinados.

La primera parte en este proceso es la recogida de datos de los *encoders* para ver la salida del sistema cuando cambia la velocidad de avance, pues se busca el modelo de comportamiento del robot cuando esto ocurre. Para ello, se fija la velocidad angular en valor nulo, y se realizan ensayos en los que se va cambiando la velocidad de avance cada cierto tiempo; en este caso cada dos segundos. En esta ocasión los ensayos realizados varían la velocidad entre 0.25 m/s y 0.75 m/s. La diferencia existe en el comienzo, pues el primero mantiene la velocidad de avance nula durante los dos primeros segundos de ensayo y el segundo lo hace durante los primeros cuatro. Se realizan estos dos ensayos porque se observa que la estabilidad es mayor cuanto más tiempo haya permanecido el vehículo en estático. En este proceso, los datos recogidos se recogen cada 10 ms, pues es el tiempo de simulación escogido.

Una vez realizados los ensayos, se guardan los valores de la velocidad angular medidos, velocidad de avance medidos, velocidad angular de referencia, velocidad de avance de referencia y el tiempo de simulación, pues se está trabajando en tiempo discreto a la hora de crear la función de transferencia más adelante.

Una vez se escoge un modelo de planta, se procede a la obtención de los valores de los parámetros. Este proceso es realizado a través de Matlab, mediante un programa basado en el método de mínimos cuadrados, utilizado con el objetivo de optimizar el resultado. Se ha partido de un código previo realizado en un laboratorio de ICAI, pero modificado porque este código estaba pensado inicialmente para un determinado robot físico con el que se trabaja en el laboratorio. En este laboratorio se trabaja en Simulink para crear la función de transferencia, para la adaptación del código previo a este proyecto se ha hecho uso de la función de *lsim()*, mediante la cual se puede construir una función de transferencia, incluso partiendo de datos discretos, como son los que se obtienen con los ensayos anteriores. Tras

varias iteraciones se obtienen los resultados para los parámetros que se buscan, además del valor de la función de coste que relaciona el valor de entrada con el valor de salida en función de lo cercanos que sean los datos. En este caso, cuando más pequeño sea, más preciso será. Para visualizarlo de una manera más clara, se representan las Figura 10, en las que se representan el valor real de salida junto con el valor de salida simulado por los parámetros obtenidos con el método de mínimos cuadrados.

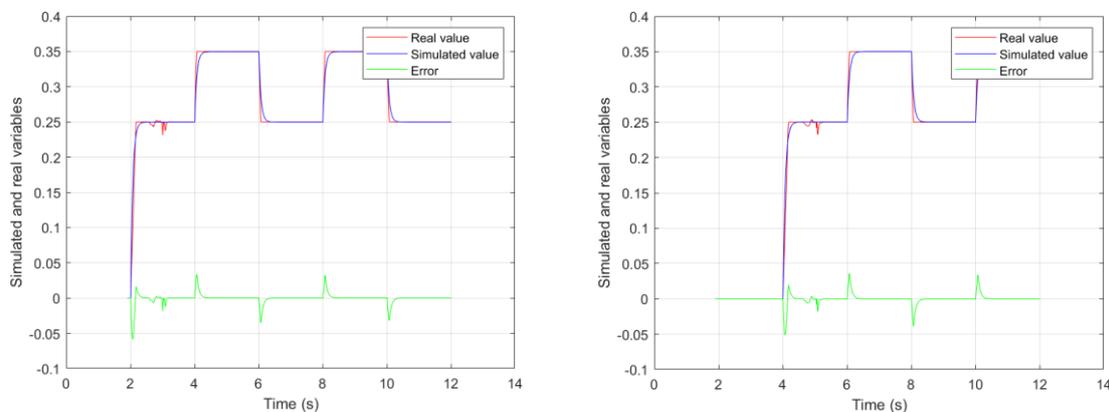


Figura 10: Gráfico comparativo de ensayos de velocidad de avance 1 y 2.

Una vez aplicado el código con los mínimos cuadrados con los datos de los ensayos, se obtuvieron los siguientes parámetros:

	$K \left(\frac{m/s}{m/s} \right)$	$T (s)$	Función de coste $\left(\frac{m/s}{m/s} \right)^2$
Primer ensayo	0.99905	0.32081	6.8336e-3
Segundo ensayo	1.0007	0.34163	6.3196e-3
Media	0.99988	0.33122	6.5766e-3

Tabla 3. Parámetros planta para velocidad de avance

Se escoge una relación media entre los parámetros de ambos ensayos, con el fin de que se acerque más a la función real. Con estos valores medios, aplicando los parámetros hallados a la Ecuación 1, la forma de la planta del modelo de navegación libre para la velocidad de avance es la siguiente:

$$P_{avance}(s) = \frac{0.99988}{0.33122 * s + 1} \approx \frac{1}{0.33 * s + 1}$$

Ecuación 2

La Ecuación 2 es utilizada como modelo matemático del robot cuando éste se desplaza en línea recta, modificando o no su velocidad de avance. Esta función de transferencia es utilizada para el diseño del control.

5.2.2 ESTUDIO DE PLANTA PARA VELOCIDAD DE GIRO

La planta para velocidad de giro es la otra de las plantas de navegación libre a modelar. Para definir esta planta, el proceso a seguir es muy similar al modelado de la planta para velocidad de avance. El primer paso es recoger los datos de los ensayos realizados, aquellos recogidos por los *encoders*. Los ensayos propuestos en esta ocasión consisten en mantener la velocidad constante a 0.5 m/s e ir variando la velocidad de giro entre 0.2 rad/s en sentido positivo y 0.2 rad/s en sentido negativo. De esta manera se puede estudiar el comportamiento del vehículo cuando trabaja con una velocidad angular cambiante. Ambos ensayos trabajan con los mismos datos, pero el primer ensayo trabaja comenzando con dos segundos sin giro alguno ($\omega=0$) y el segundo ensayo trabaja con un comienzo nulo de cuatro segundos. Los valores de velocidad de avance y giro medidos, y valores de velocidad de avance y giro de referencia son recogidos, además del tiempo de simulación con el que se trabaja, que es otra vez de 10 ms.

Una vez se han guardado los datos de los ensayos, se procede al análisis de mínimos cuadrados. Una vez realizados ambos análisis, se puede visualizar y garantizar la validez de la planta modelada mediante una gráfica que junte la salida real y la salida simulada. Cada

uno de los ensayos y la comparativa entre valor real y simulación quedan representados en la Figura 11.

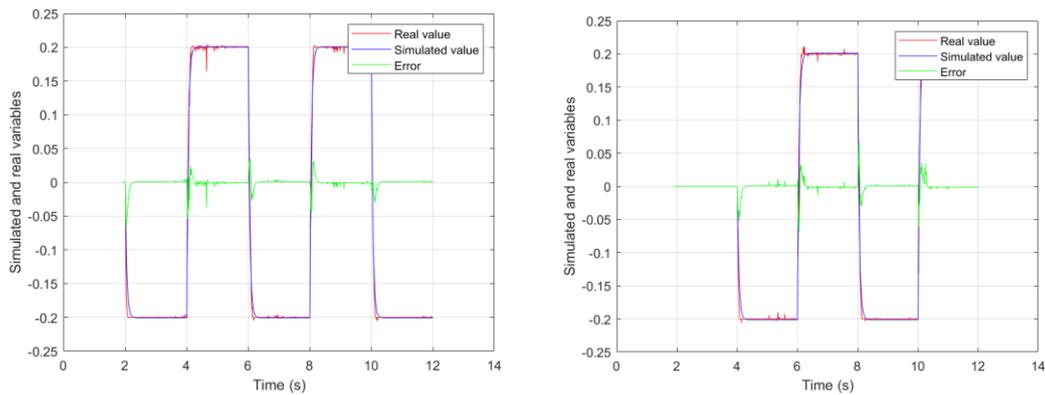


Figura 11: Gráfico comparativo de ensayos de velocidad de giro 1 y 2

Una vez aplicado el código con los mínimos cuadrados con los datos de los ensayos, se obtuvieron los siguientes parámetros:

	$K \left(\frac{m/s}{rad/s} \right)$	$T (s)$	$Función\ de\ coste$ $\left(\frac{rad/s}{m/s} \right)^2$
Primer ensayo	1.0033	0.23785	7.3393e-3
Segundo ensayo	1.0061	0.25229	7.5280e-3
Media	1.0047	0.24507	7.4337e-3

Tabla 4. Parámetros planta para velocidad de giro

Por tanto, y aplicando los parámetros medios a la Ecuación 1, la forma de la planta del modelo de navegación libre para la velocidad de giro es la siguiente:

$$P_{giro}(s) = \frac{1.0047}{0.24507 * s + 1} \approx \frac{1}{0.245 * s + 1}$$

Ecuación 3

Esta última función de transferencia, la Ecuación 3, es la utilizada para el diseño del control de velocidad de giro.

5.3 PLANTA PARA EL MODELO DE NAVEGACIÓN DE SEGUIMIENTO DE PARED

Este modelo de planta es bastante más complejo que los otros dos anteriores. Con el objetivo de explicarlo de manera más visual, se añade la ilustración:

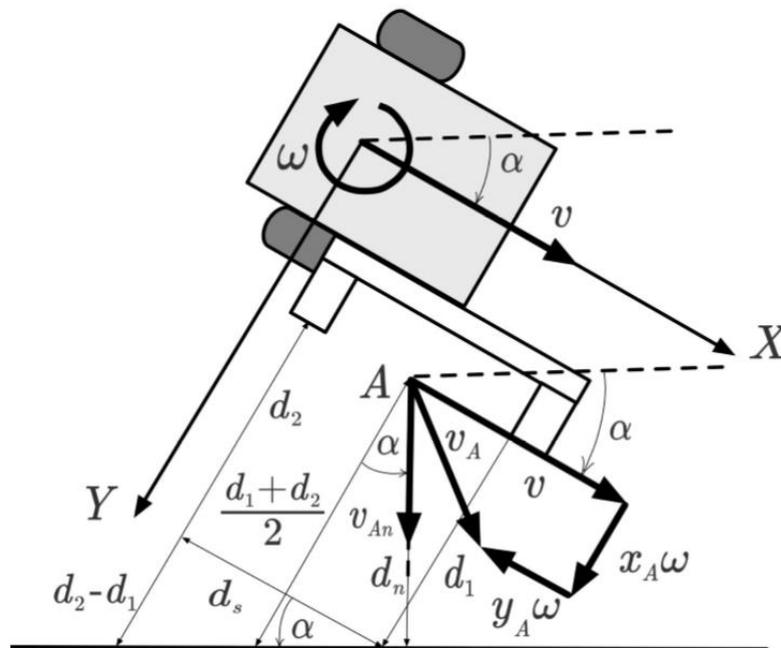


Figura 12: Explicación de las distancias y ángulos medidos según los sensores utilizados [14]

Aunque la Figura 12 represente la imagen de un robot que consta de exclusivamente dos sensores y el robot a utilizar conste de 16, puesto que solo se trabaja con dos de ellos, el modelo plasmado es suficientemente similar.

Es importante entender cómo se calculan cada una de las medidas en el gráfico y lo que significan. Por ejemplo:

$$\alpha = \text{atan}\left(\frac{d_2 - d_1}{d_s}\right)$$

$$d_n = \frac{d_1 + d_2}{2} * \cos(\alpha)$$

Siendo α el ángulo de orientación del vehículo con la pared y d_n la proyección normal de la distancia media medida por los sensores. Por otro lado, el punto A es el punto medio entre ambos sensores, por lo que será el punto de trabajo a la hora de analizar la planta.

Analizando la velocidad del dispositivo, se podría dividir en dos según los ejes x e y :

$$v_{Ax} = v - y_A * \omega$$

$$v_{Ay} = x_A * \omega$$

Por tanto, la v_{An} en función del tiempo tiene como forma:

$$v_{An}(t) = v_{Ax} * \text{sen}(\alpha) + v_{Ay} * \text{cos}(\alpha)$$

Si se analiza la proyección de la distancia media medida por los sensores en función del tiempo se obtiene:

$$d_n(t) = d_n(0) - \int_0^t v_{An}(t) dt$$

Además de todas esas ecuaciones, aparece una que relaciona la tensión diferencial con la velocidad de giro del vehículo:

$$\frac{d\omega}{dt} = \frac{1}{T_m} * \omega + \frac{K_m}{T_m} * u_d$$

Y observando las ecuaciones con las que se pudiese trabajar, aparece la no linealidad de la función de v_{An} . Después de aplicar las matemáticas necesarias para la linealización, que cuyo cálculo no procede en este proyecto se llega al siguiente sistema de ecuaciones:

$$\frac{d}{dt} \begin{bmatrix} \Delta\omega \\ \Delta\alpha \\ \Delta d_n \end{bmatrix} = \overbrace{\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{T_m} & 0 & 0 \\ 1 & 0 & 0 \\ -x_A & -v_o & 0 \end{bmatrix}}^A \begin{bmatrix} \Delta\omega \\ \Delta\alpha \\ \Delta d_n \end{bmatrix} + \overbrace{\begin{bmatrix} \frac{K_m}{T_m} & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}}^B \begin{bmatrix} \Delta u_d \\ \Delta\omega_{pared} \end{bmatrix}$$

Figura 13: Modelo linealizado planta de seguimiento a pared

Por último, se obtiene la función de transferencia entre la componente diferencial del motor y la distancia a la pared, que en esta ocasión será entre la velocidad de giro y la distancia a la pared, pues son proporcionales. Esto da como resultado de planta una planta compuesta por dos. En la Ecuación 4 se observan las dos plantas que conforman el modelo de la planta de seguimiento a pared. La primera de ellas relaciona la velocidad de giro con el ángulo respecto a la pared y la segunda relaciona el ángulo a la pared con la distancia a ésta.

$$P_{pared}(s) = \frac{\frac{P_1(s)}{K_m}}{(1 + T_m * s) * s} * \frac{\frac{P_2(s)}{-v_o * \left(1 + \frac{x_A}{v_o}\right)}}{s} = \frac{-v_o * K_m * \left(1 + \frac{x_A}{v_o} * s\right)}{(T_m * s + 1) * s^2}$$

Ecuación 4

Como se ha explicado anteriormente, al punto A hace referencia al valor intermedio de los sensores. Teniendo en cuenta que para este proyecto los sensores escogidos son los sensores clasificados como sensor 8 y sensor 9, por sus características de perpendicularidad con el vehículo y entre ellos:

$$\begin{aligned}x_8 &= 1.0641e - 1 \text{ m} \\x_9 &= -1.1029e - 1 \text{ m} \\x_A &= -3.88e - 3 \text{ m}\end{aligned}$$

Por otro lado, el valor de v_0 está relacionado con la velocidad a la que vaya a estar comúnmente desplazándose el vehículo, por lo que el valor escogido será $v_0 = 0.5 \text{ m/s}$. Los valores de K_m y T_m son los mismos que los calculados en la planta de velocidad de giro. Por tanto, siguiendo la forma de la Ecuación 4, se obtiene la siguiente función de transferencia como planta:

$$P_{pared}(s) = \frac{\overbrace{1}^{P_1(s)}}{(1 + 0.245 * s) * s} * \frac{\overbrace{-0.5 * \left(1 + \frac{-3.88e - 3}{0.5} * s\right)}^{P_2(s)}}{s}$$

$$P_{pared}(s) = \frac{-0.5 * 1 * \left(1 + \frac{-3.88e - 3}{0.5} * s\right)}{(0.245 * s + 1) * s^2}$$

Con esta planta se procede al cálculo del control de seguimiento de pared.

Capítulo 6. CONTROL PID

6.1 CONTROL PID COMO CONCEPTO

Definido como control clásico, el control PID (control proporcional, integral y derivativo) es uno de los controles más aplicados en los sistemas industriales, siendo así que el 95% de estos controles son controles PID, siendo la mayoría de ellos controles PI, pues se caracterizan por lograr una diferencia entre referencia y salida de cero, es decir con error nulo [15].

Este control se caracteriza por tener la capacidad de disminuir el tiempo de establecimiento a la par de disminuir el error de establecimiento. Es por esto por lo que cuenta con la acción derivativa y la integral.

La acción proporcional debe ser estudiada teniendo en cuenta las posibles perturbaciones que pueden modificar la salida. Para esto, sería necesario analizar dónde están apareciendo estas perturbaciones, pues si éstas aparecen como mediciones de salida cuanto mayor sea la ganancia proporcional, mayor será el error con la referencia. Sin embargo, al hallarse la perturbación entre control y sistema (perturbación de carga) el error será menor cuanto mayor sea la ganancia estática [15].

La acción derivativa tiene como objetivo disminuir el tiempo de establecimiento de la salida. Esto es conseguir que la señal llegue a un establecimiento estático de manera más rápida de lo que se haría con un único control proporcional.

Por último, la acción integral permite encontrar un error entre salida y referencia que sea cero o por lo menos lo más pequeño posible. La acción integral no puede ser empleada en todas las plantas, pues si la planta para la que se está diseñando un control contiene de por sí la acción integral, el hecho de añadir un control integral sólo desestabilizaría al sistema.

Es la composición de estas tres acciones la que da lugar a un control PID, que consiste en el control preciso, rápido y sin error que sea lo más adaptado a la planta del sistema.

6.2 FORMA DEL PID UTILIZADO

En este proyecto se ha decidido utilizar la forma de la Ecuación 5 como diseño del control utilizado en las diferentes plantas a controlar:

$$C(s) = K_p * \frac{1 + I * s}{I * s} * \frac{1 + D * s}{1 + f * D * s}$$

Ecuación 5

Donde K_p hace referencia a la ganancia proporcional estática, I al factor de integración, D al factor de derivación y f es el factor de filtrado.

A la hora del diseño se deben tener en cuenta las siguientes relaciones:

$$K_I = \frac{K_p}{I}$$

$$\varphi_{PD} = \varphi_C - \varphi_{PI}$$

$$f_M(\varphi_{PD}) \geq f \geq 0$$

La primera de todas hace referencia a la ganancia de integración. Cuanto mayor sea este valor, más preciso será el sistema. El margen de fase del control derivativo está relacionado con el margen de fase del control total y con el margen que se le aplique a la acción integral. También se debe tener en cuenta que el factor de filtrado no puede tener un valor negativo, y que nunca debe superar al factor de filtrado de margen de fase máximo para el derivativo.

Entre todos estos, los principales parámetros que se deben tener en cuenta a la hora de diseñar controles PID son:

- Margen de fase: es el retraso de fase que habría que añadir al lazo abierto para obtener un sistema oscilante. Al evaluarlo sobre el diagrama de Black, aumentar o retrasar la fase sería equivalente a desplazar la curva hacia la izquierda, mientras que Nyquist sería rotar la curva en sentido horario [16]. A la hora de evaluarlo en el control, cuanto mayor sea el margen de fase, menos sobrepaso habrá en la curva, pero es probable que de esta manera se aumente el tiempo de establecimiento.
- Margen de ganancia: es el factor por el que habría que multiplicar el lazo abierto para obtener un sistema oscilante. Al aumentar la ganancia, se aumenta la curva en el diagrama de Nyquist y se traslada verticalmente hacia arriba en el diagrama de Black. A la hora de diseñar, en este proyecto no se tendrá en cuenta, pues se diseña a partir del margen de fase [16].

Es importante tener en cuenta que hay dos grados de libertad a la hora de diseñar el control PID, por lo que no existe una única opción válida. Debe ser realizado de manera manual teniendo en cuenta los parámetros previamente explicados.

Una vez teniendo claro los parámetros a estudiar, el control más acertado posible debería seguir las siguientes características:

- Estabilidad: indica el comportamiento inicial y en régimen permanente del sistema, es decir, si es oscilante o no. Existen diferentes métodos para estudiar la estabilidad de un sistema, entre los que destacan el criterio del reverso, con la ayuda de los gráficos de Nyquist y Black (Figura 14), el criterio de las raíces o el criterio de Routh-Hurwitz. Una ampliación del criterio del reverso sería el criterio de Nyquist [16].

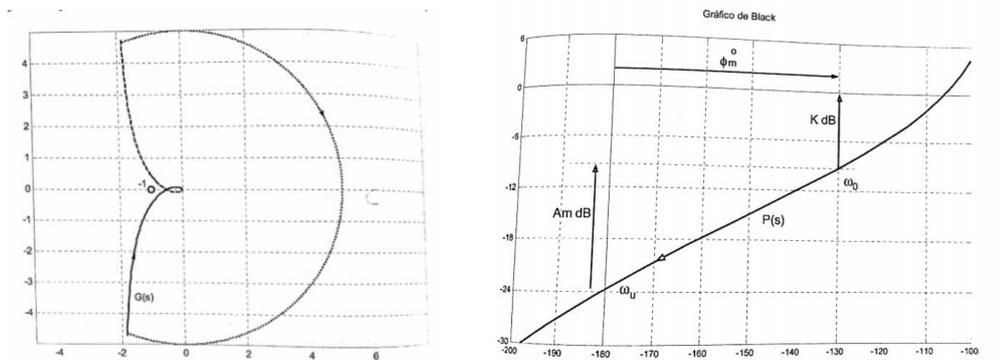


Figura 14: Diagramas de Nyquist y Black [16]

- Amortiguamiento: trata de explicar la facilidad con la que un sistema llega a establecerse teniendo en cuenta también el pico que pueda haber. Los parámetros que se encargan de regular esta característica son principalmente la A_m y ϕ_m [16].
- Rapidez: hace referencia al tiempo que tarda en establecerse. La rapidez se suele hacer como comparativa en sistemas de mismo amortiguamiento. Los parámetros a tener en cuenta son ω_0 , ω_u , ω_s y ω_c [16].
- Precisión en régimen permanente: error debido a la diferencia entre referencia y salida. Dependiendo de la ganancia proporcional, este error será mayor de ser una K_p alta a serlo baja [16].

Aunque el diseño del control se realice de manera que derivación e integración interactúen, a la hora de implantar éste mismo se hace de manera no interactiva. Por tanto, los parámetros que aparecen ahora son T_d , T_i y K , dos constantes de tiempo y una ganancia. La forma de actuar es la representada en la Figura 15, con acción diferencial aplicada a la salida. Un buen resumen para explicar el comportamiento del control según cómo varían K , T_d y T_i sería el expuesto en la Tabla 5:

	K aumenta	T_i disminuye	T_d aumenta
Estabilidad	Se reduce	Disminuye	Aumenta
Velocidad	Aumenta	Aumenta	Aumenta
Error estacionario	No eliminado	Eliminado	No eliminado
Área de error	Se reduce	Disminuye hasta cierto punto	Se reduce
Perturbación control	Aumenta bruscamente	Aumenta gradualmente	Aumenta bruscamente
Frecuencia lazo	No afecta hasta cierto punto	Disminuye	Aumenta

Tabla 5: Recurso explicativo del comportamiento del PID según sus parámetros

Por último y mediante una serie de relaciones que existen entre I y T_i , D y T_d y f y N , se llega a la Ecuación 6, utilizada para la implementación de cada control PID.

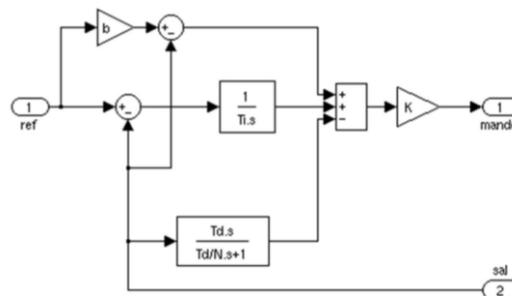


Figura 15: Control PID no interactivo con 2 grados de libertad [14]

$$C(s) = K * \left(b + \frac{T_d * s}{1 + \frac{T_d}{N * s}} + \frac{1}{T_i * s} \right)$$

Ecuación 6

6.3 DISEÑO DEL PID

Una vez decidido el modelo del control a emplear, se procede al diseño de este mismo. En este proyecto, debido a la forma de las plantas, los controles de navegación libre siguen un proceso muy similar al diseñar, pero el control de seguimiento de pared sigue otro planteamiento.

A continuación, se presentan los controles más lógicos planteados para más adelante elegir la solución más apropiada acorde con los resultados.

6.3.1 CONTROL PID PARA PLANTA DE VELOCIDAD DE AVANCE

Para explicar cada decisión tomada a la hora de diseñar el control de planta de velocidad de avance, primero hay que recordar la forma de la planta previamente hallada. En la Ecuación 2 se mostraba el modelo de la planta hallado por el método de mínimos cuadrados. Este es el modelo utilizado para hallar el control.

Con el objetivo de obtener los valores de los parámetros, se procede a utilizar un programa creado para el diseño de control PID. El código de Matlab ‘GUI.control.digital.m’ creado por Juan Luis Zamora, posibilita mediante la modificación de ciertos parámetros, la construcción de gráficos como el de Black o la respuesta a un escalón de manera casi simultánea a la modificación de ciertos valores. En el programa se introducen los datos de la planta y el tiempo de simulación (en este caso sería los 10 ms utilizados previamente para hallar la planta).

Debido a que la estructura de la planta no contiene ninguna integración y a que la respuesta es relativamente rápida, los controles propuestos para estudiar serán: control proporcional (P), control proporcional integral (PI) y control proporcional con acción integral y diferencial (PID).

6.3.1.1 Diseño del control P

En este tipo de control el único parámetro modificable es el margen de fase. Con este parámetro se consigue amortiguar la señal, pero siempre se debe tener en cuenta que al amortiguar la señal demasiado se puede volver muy lento el sistema. Por eso se busca un punto intermedio obteniendo así el siguiente resultado:

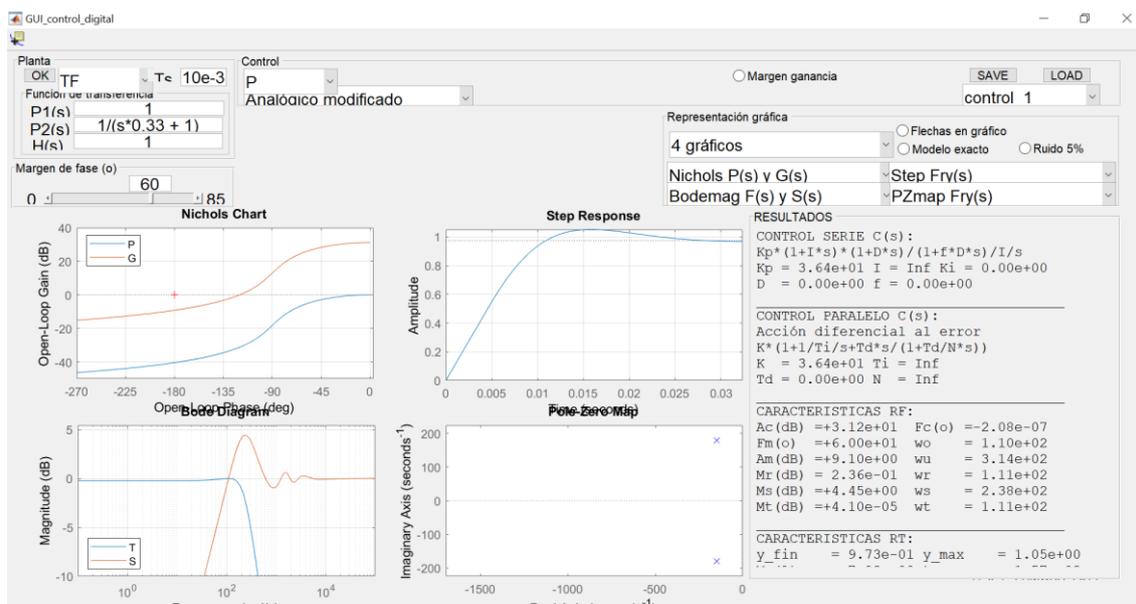


Figura 16: Resultado del control P para la planta de velocidad de avance

Se considera que es un buen control puesto que se estabiliza al llegar a un valor muy cercano a 1. Además, el valor de salida máximo no sobrepasa el 15% y el tiempo de establecimiento está en torno a los 30 ms. Atendiendo al diagrama de Nichols, puesto que corta con la línea del -180° en un valor negativo de los decibelios, se considera un sistema estable.

Por tanto, los parámetros que se implementan son los siguientes:

K	T_d	T_i	b	N
36.4	0	Inf	1	Inf

Tabla 6: Parámetros del control P para la planta de velocidad de avance

6.3.1.2 Diseño del control PI

En esta ocasión se trabaja con más parámetros a modificar que en el caso del control proporcional. Ahora se deberán tener en cuenta los factores de margen de fase, velocidad y ponderación de la referencia. Se comienza buscando un amortiguamiento acertado, y se establece el margen de fase en 60° . El resultado obtenido no es malo, pero es mejorable. El pico de la salida alcanza un sobrepaso del 6%, que es reducible modificando levemente la ponderación de la referencia, pues al disminuirla también lo hace el pico. Por tanto, el resultado final es el de la Figura 17:

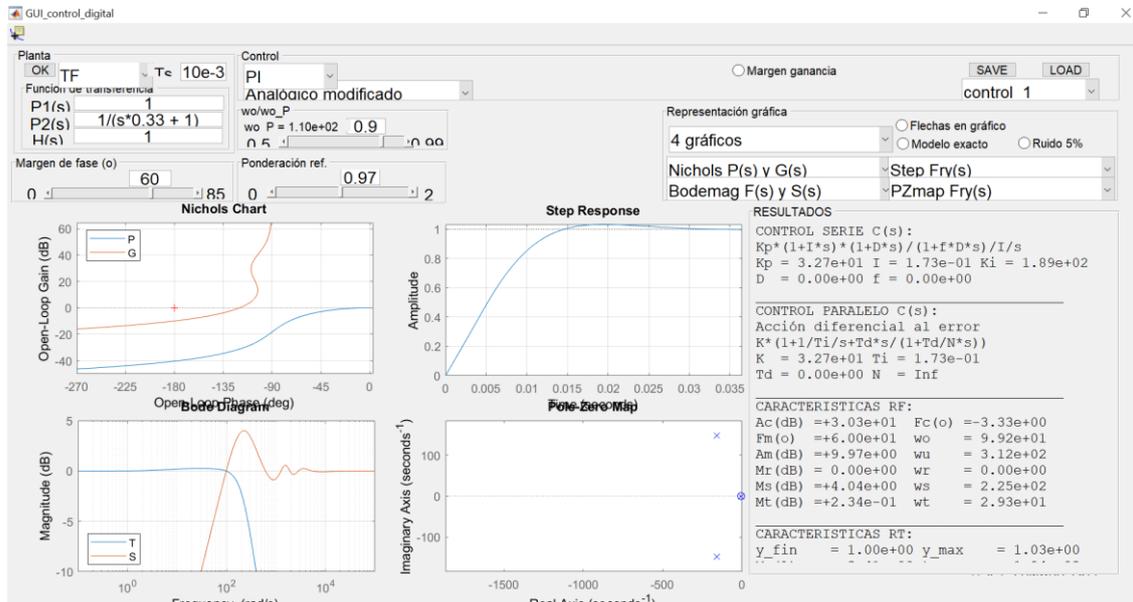


Figura 17: Resultado del control PI para la planta de velocidad de avance

Los parámetros que se utilizan en el control PI son, por tanto:

K	T _a	T _i	b	N
32.7	0	0.173	0.97	Inf

Tabla 7: Parámetros del control PI para la planta de velocidad de avance

6.3.1.3 Diseño del control PID

Puesto que a la hora de diseñar el control PID existen muchos parámetros a modificar, se comienza con el margen de fase para amortiguar la salida; en este caso se amortigua con 60° de margen de fase. De esta manera se consigue amortiguar la salida, aunque sigue siendo superior al 20%, además de no establecerse antes de los 0.1 s. Con el objetivo de aumentar la ganancia integral se modifica el retraso de fase aumentándolo de 10° a 15°. Es entonces

cuando aumenta tanto la ganancia integral como el pico a la salida. Para no modificar la ganancia integral se decide modificar la ponderación de la referencia, que se disminuye de 1 a 0.8, disminuyendo así el pico a un porcentaje inferior al 20%. Velocidad y factor de filtrado se deciden no modificar, pues no se obtienen ningún resultado atractivo a la hora de hacerlo. El resultado del control es el expuesto en la Figura 18:

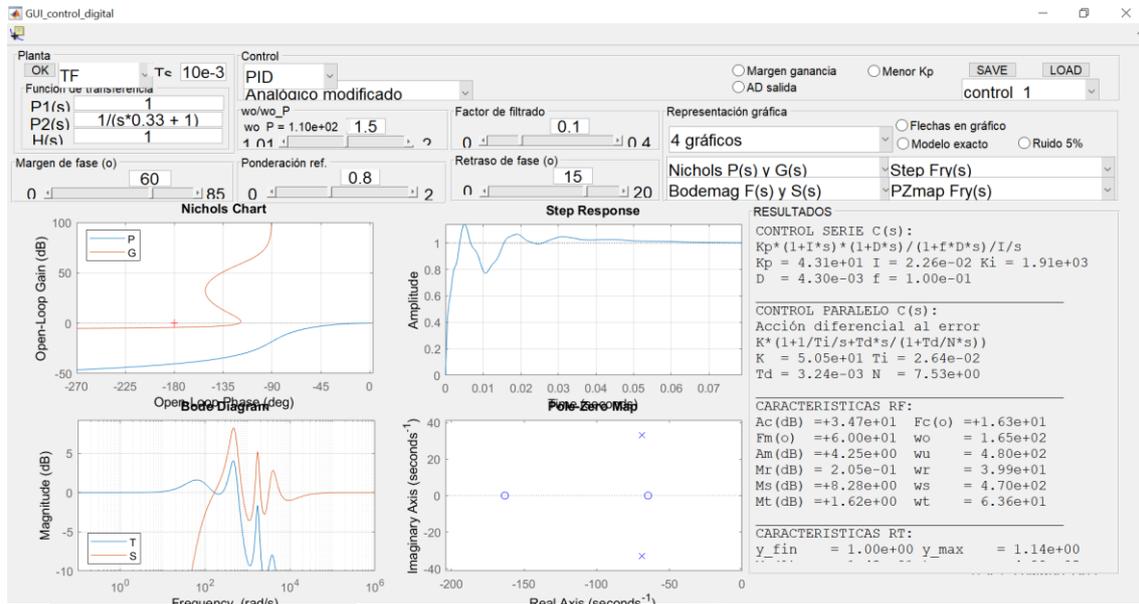


Figura 18: Resultado del control PID para la planta de velocidad de avance

Los parámetros por tanto que se utilizarán para simular este control serán:

K	T_d	T_i	b	N
50.5	0.00324	0.0264	0.8	7.53

Tabla 8: Parámetros del control PID para la planta de velocidad de avance

6.3.2 CONTROL PID PARA PLANTA DE VELOCIDAD DE GIRO

Las plantas presentan un gran parecido de estructura y de valores, por lo que se comprueba si el control calculado para la planta de avance es suficiente para este otro. Al comprobar que no es válido para esta otra planta, se repite el proceso anterior para la planta de velocidad de giro, que utilizará como función de transferencia para los cálculos los de la Ecuación 3.

6.3.2.1 Diseño del control P

Se procede a seguir los pasos de diseño realizados para la otra planta; en el control P el único parámetro a modificar será el margen de fase.

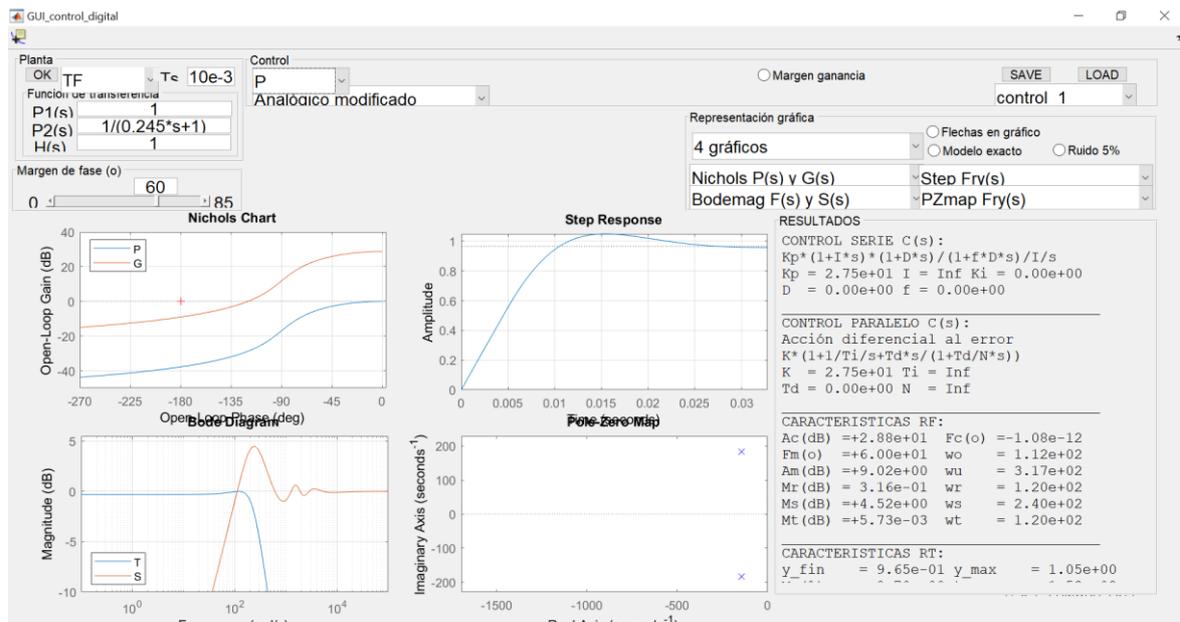


Figura 19: Resultado del control P para la planta de velocidad de giro

Observando la semejanza que tiene con el control anterior se entiende que sea bastante preciso. Por tanto, los parámetros que se implementan son los siguientes:

K	T _d	T _i	b	N
27.5	0	Inf	1	Inf

Tabla 9: Parámetros del control P para la planta de velocidad de giro

6.3.2.2 Diseño del control PI

A la hora de diseñar este control aparecen más parámetros a modificar, pero se comienza por el margen de fase, que es aumentado levemente para amortiguar un poco la salida, pero sin hacerla demasiado lenta. La ponderación a la referencia también se modifica levemente para que se disminuya el pico de la salida. El resultado es el de la Figura 20:

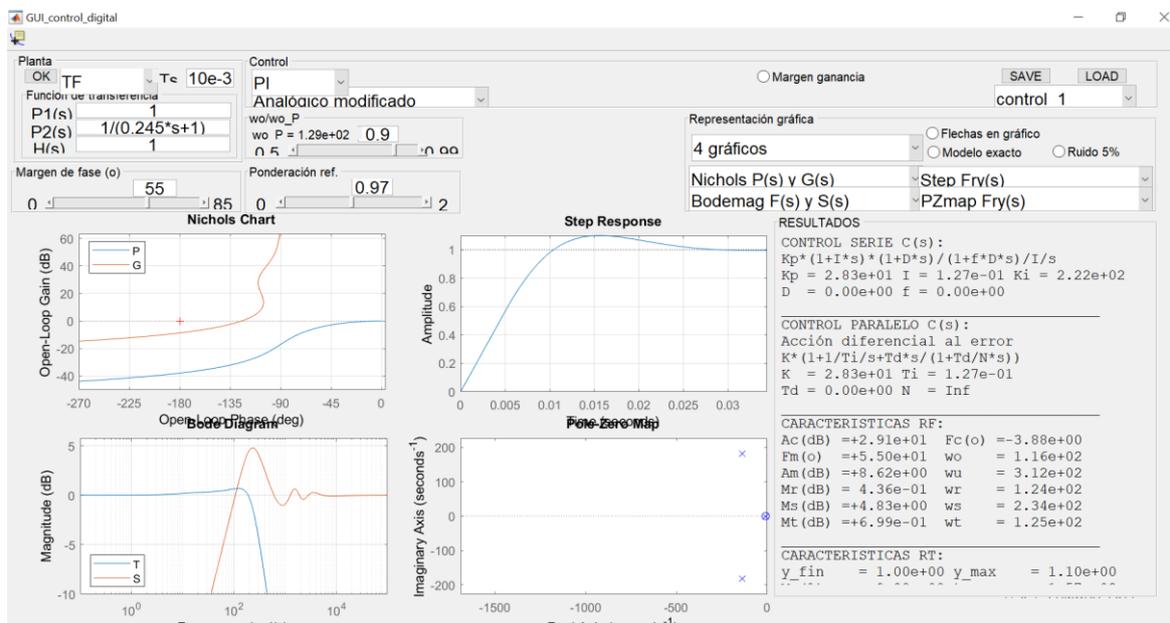


Figura 20: Resultado del control PI para la planta de velocidad de giro

Los parámetros que se utilizan en el control PI son, por tanto:

K	T_d	T_i	b	N
28.3	0	0.127	0.97	Inf

Tabla 10: Parámetros del control PI para la planta de velocidad de giro

6.3.2.3 Diseño del control PID

Siguiendo el mismo orden que en el resto de los controles, se decide modificar el margen de fase en primer lugar para que exista una amortiguación relativamente apropiada y que a la hora de modificar el resto de los factores ésta no empeore. Debido a la extraña señal de salida, el amortiguamiento se debe agrandar, con un margen e fase de 60°. También aumenta el retraso de fase y disminuye la ponderación a la referencia, con el objetivo de que el pico disminuya. Se disminuye el factor de filtrado a la mitad, es decir de 0.1 a 0.05, para aumentar un poco la ganancia de la integración y hacer al sistema un poco más preciso. El resultado final es el de la Figura 21:

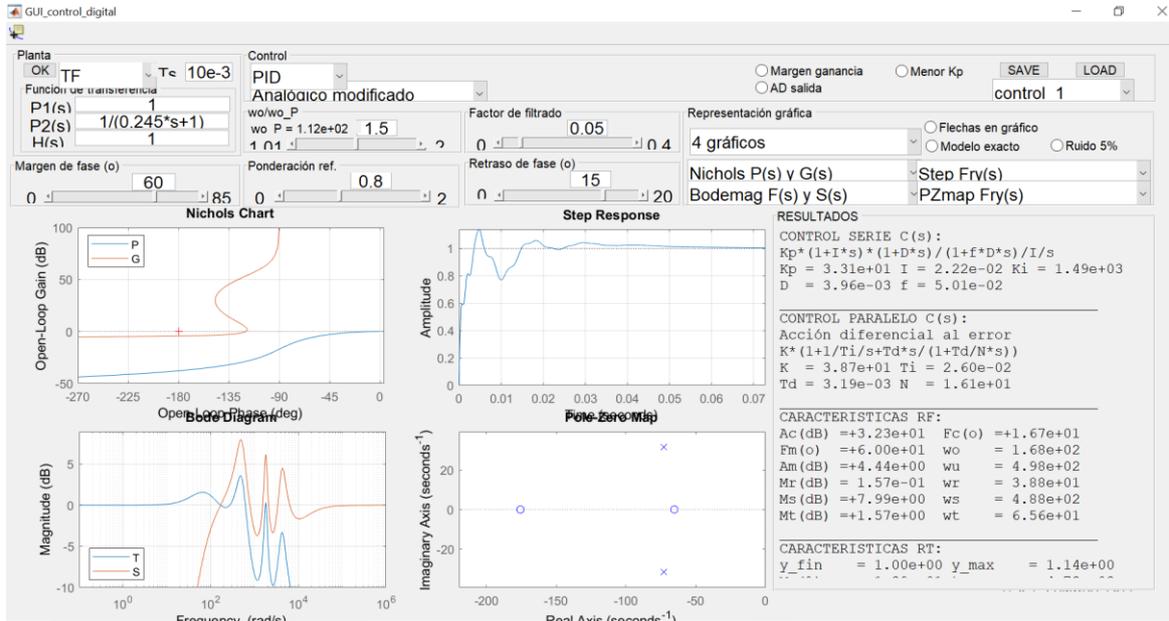


Figura 21: Resultado del control PID para la planta de velocidad de giro

El resultado de los parámetros es el siguiente:

K	T_d	T_i	b	N
38.7	0.00319	0.026	0.8	16.1

Tabla 11: Parámetros del control PID para la planta de velocidad de giro

6.3.3 CONTROL PID PARA PLANTA DE MODELO DE SEGUIMIENTO DE PARED

La forma de la planta de modelo de seguimiento a pared es un poco más compleja de lo que eran las dos anteriores, puesto que tiene una doble integración. Recordando su forma presentada en la Ecuación 5:

$$P_{pared}(s) = \frac{\frac{P_1(s)}{0.5}}{(1 + 0.245 * s) * s} * \frac{\overbrace{-0.5 * \left(1 + \frac{3.88e - 3}{0.5}\right)}^{P_2(s)}}{s}$$

La planta está compuesta por dos plantas. La primera de ellas transforma la velocidad en ángulo de posicionamiento del robot y la segunda ese ángulo lo transforma en distancia a la pared. Aplicar un control PID clásico a esta planta con integraciones sólo complicaría la situación. Es por esto y debido a que se puede medir el ángulo de giro a partir de los sensores ultrasónicos que están colocados sobre el vehículo, se decide aplicar un control en cascada.

Este control no ha sido explicado con anterioridad, pero consiste en hacer controles anidados según las plantas que haya. Un primer control es aplicado sobre una de las plantas y el siguiente es aplicado sobre ambas plantas, incluyendo al primer control. En la Figura 22 queda representado la forma del control en cascada, donde P_1 y P_2 :

$$P_1(s) = \frac{K_m}{(T_m * s + 1) * s} = \frac{0.5}{(0.245 * s + 1) * s}$$

$$P_2(s) = \frac{-v_0 * \left(1 + \frac{x_A}{v_0}\right)}{s} = \frac{-0.5 * \left(1 + \frac{-3.88e - 3}{0.5}\right)}{s}$$

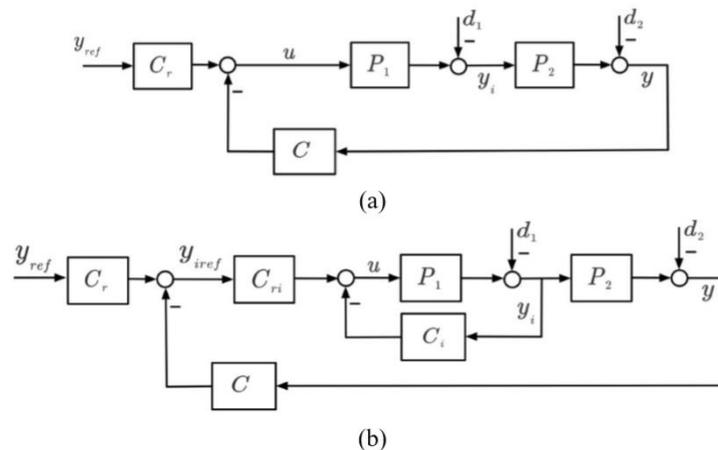


Figura 22: Control de un único lazo (a) y control en cascada (b)[14]

Una vez se ha decidido que se quiere trabajar con un control en cascada, se debe tomar la decisión de qué pruebas se realizarán para los controles. Una vez visto que las plantas poseen integraciones, se descartan las opciones de que aparezcan control PI y PID en el control en cascada. Por tanto, las dos opciones escogidas serán:

- Realizar un control P en el lazo interno y un control PD en el lazo externo.
- Realizar un control PD en el lazo interno y un control P en el lazo externo.

6.3.3.1 Control PD en lazo interno y control P en lazo externo

Este control tendrá mayor rapidez a la hora de medir el ángulo con la pared. Para el diseño de este control es necesaria la ayuda de un script de Matlab CONTROL_YAW_ANG_PID_TR.m, que trata de diseñar por asignación de polos. La dinámica a la que se quiere proceder es el diseño del control PD mediante asignación de polos y diseño del control P mediante respuesta en frecuencia, como se ha hecho con anterioridad con el código de GUI_control_digital.m.

Se procede a asignar los valores de la planta en el código

```
1. Km = -1;  
2. Tm = 0.245;  
3. P = Km / (Tm*s+1)/s; % Car  
4. % Planta con retardo por discretización  
5. ts = 0.01;  
6. Pam = P;  
7. Psimp = Km / (Tm*s+1)/s;  
8. Pd = 0;
```

Observando la función de transferencia obtenida entre referencia y salida al aplicar el control, aparecen tres polos y un cero. Se busca hacer que uno de los polos anule al cero, pues su efecto es pequeño ya que es bastante grande, es decir, lejano al cero. Buscando una $\omega_{PD}=2* \omega_P$, se consigue la siguiente función de transferencia:

$$Fry_{pd} = \frac{16.66 * (s + 40.82)}{(s + 35.61) * (s + 6.217) * (s + 3.072)} \approx \frac{16.66}{(s + 6.217) * (s + 3.072)}$$

El cero de -40.82 es semejante al polo de -35.61, por lo que se decide trabajar sin ellos para simplificar el control proporcional a obtener a continuación. Una vez elegido este como la función de transferencia de la primera planta una vez se ha aplicado el control, se sabe que el control PD elegido para el lazo interno es de las siguientes características:

K_{PD}	-4.0816
T_d	0.245
b	1
N	10

Tabla 12: Parámetros control PD interno del control en cascada

Por lo tanto, la forma de este control será:

$$C_{PD} = -4.0816 * \left(1 + \frac{0.245 * s}{1 + \frac{0.245 * s}{10}} \right)$$

A continuación, se procede a visualizar la función de transferencia que queda como resultado de aplicar el control en la planta 1 y juntarlo con la planta 2. Para ello se utiliza Matlab para operar:

```

1. %% CONTROL EN CASCADA PLANTA DE SEGUIMIENTO A PARED
2. % Primero con PD en lazo cerrado y P en lazo externo
3. load TF_YAW_ANG_PID_TR_pd
4. v_o = 0.5
5. x_A = -0.003
6. s = tf('s');
7. zpk(Fry_pd)
8. MODEL_YAW_ANG_CL = tf(Fry_p);

```

```

9. MODEL WALL_DIST_CC = v_o*...
10. (1+x_A/v_o*s)...
11. /s*MODEL_YAW_ANG_CL;

```

Como resultado se obtiene:

$$F_{ry} = \frac{-0.05 * s + 8.33}{s^3 + 9.289 * s^2 + 19.1 * s}$$

Con esta función de transferencia se trabaja con respuesta en frecuencia para hallar el control P óptimo para el control en cascada que se quiere diseñar. El único parámetro posible modificable es el margen de fase, por lo que se escoge un margen de fase que a la vez que amortigüe la señal, no la retrase demasiado. El resultado obtenido es el de la figura con una ganancia $K = 3.24$.

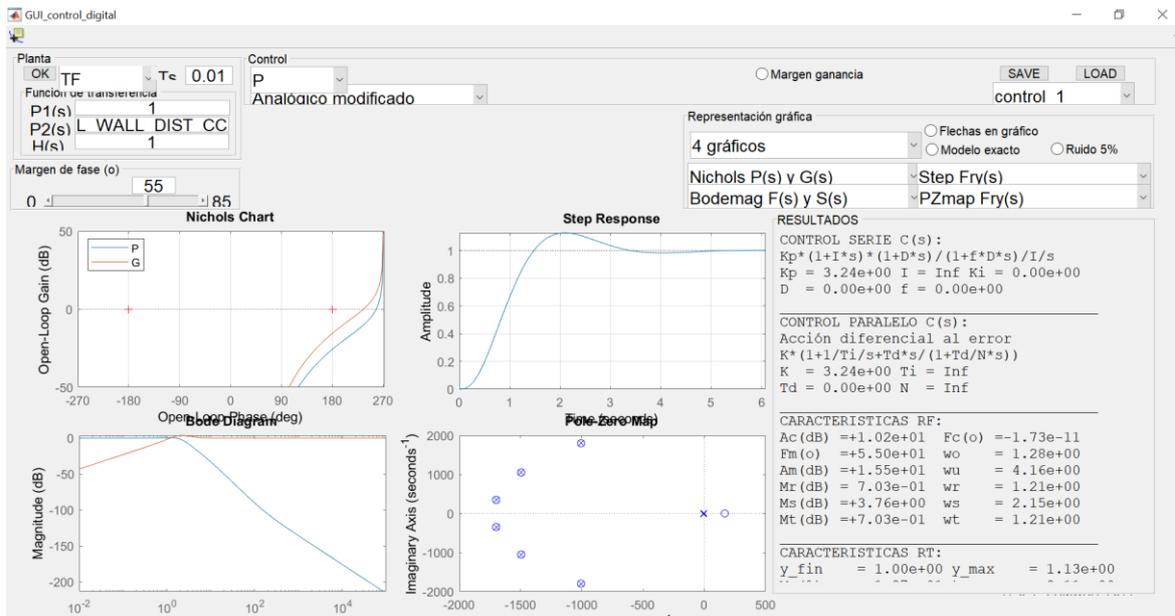


Figura 23: Diseño final por frecuencia del control P externo del control cascada.

6.3.3.2 Control P en lazo interno y control PD en lazo externo

El proceso que seguir en este control es el mismo que para el anterior. Primero se diseña según los polos, pero en esta ocasión se diseña un control P. Al trabajar con el control se obtiene una función de transferencia entre referencia y salida:

$$Fry_p = \frac{4.165}{s^2 + 4.082 * s + 4.165}$$

Para este control, la ganancia es $K=-1.0204$ y el amortiguamiento es crítico (igual a 1), para que no haya rebotes a una vez se establezca la señal.

Se realizan las operaciones necesarias para agrupar las dos plantas y el control interno en una sola función de transferencia con la que operar más adelante y se obtiene:

$$Fry = \frac{-0.01249 * s + 2.082}{s^3 + 4.082 * s^2 + 4.165 * s}$$

Utilizando esta función de transferencia para el diseño del control PD externo, una vez modificados los parámetros apropiados, se obtienen los resultados de la Figura 24:

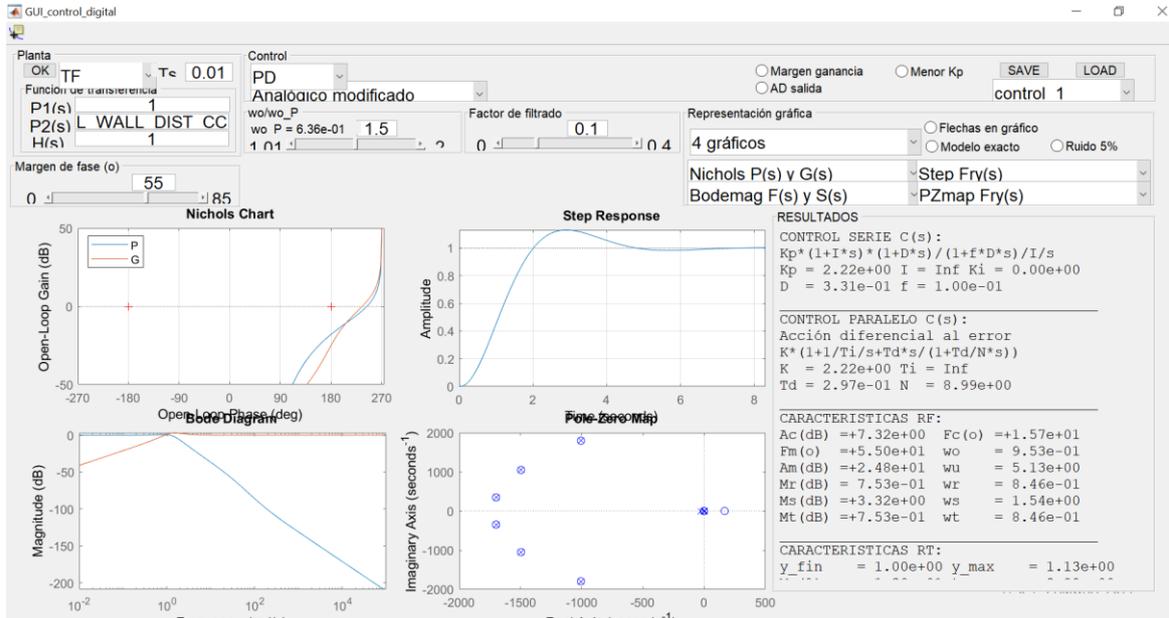


Figura 24: Diseño final por frecuencia del control PD externo del control cascada.

Por tanto, los parámetros para el control externo son:

K_{PD}	2.22
T_d	0.297
b	1
N	9

Tabla 13: Parámetros control P interno del control en cascada

Capítulo 7. CONTROL PREDICTIVO

7.1 CONTROL PREDICTIVO COMO CONCEPTO

El control predictivo aparece en primera instancia en una compañía de refinería de petróleo, OilShell. El objetivo de este control en el año 1979 cuando se desarrolló por primera vez fue el de buscar una solución a un problema con planta multivariante y con entradas y salidas no medidas [17].

El control predictivo, por tanto, cumple con el objetivo de completar el algoritmo con suposiciones de “lo que está por venir”. Al igual que le ocurre al PID, este control tiene en cuenta los sucesos pasados y presentes, pero también posee cierta inteligencia para prevenir lo que se acerca y cambiar su comportamiento en función de eso. El control predictivo debe ser aplicado cuando el cambio con respecto al control PID es notable, pues la carga computacional del primero es mucho mayor que la del segundo.

A la hora de diseñar el algoritmo del control predictivo, se tienen en cuenta conceptos muy intuitivos y además, sus salidas no dan grandes saltos, pues siempre se tienen que establecer en una banda de operación. En la Figura 25: Diagrama de bloques de un control predictivo se muestra el esquema general del control predictivo.

El control predictivo, se ocupa de una parte más lenta del proceso, pues trabaja en función a las variables de producción y de calidad, no como trabaja el PID que se encarga de trabajar según las variables físicas, que son obtenidas de manera más instantánea [17].

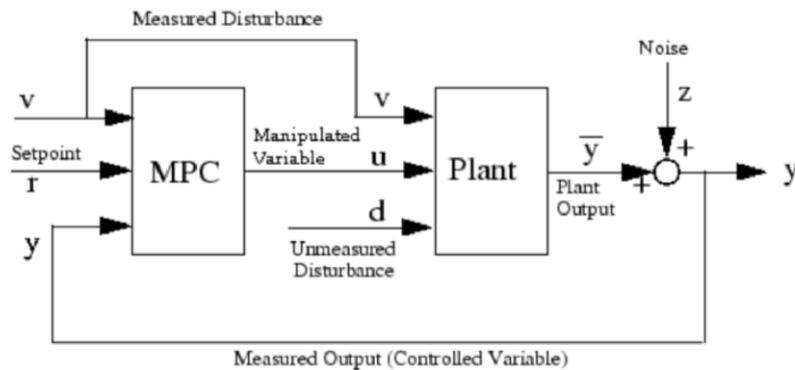


Figura 25: Diagrama de bloques de un control predictivo

7.2 FUNCIONAMIENTO Y DISEÑO DEL CONTROL PREDICTIVO UTILIZADO

El control predictivo se compone de muchas variables de diferentes formas, por lo que para explicar el funcionamiento de éste, es necesario señalar primero los tipos de variables que pueden aparecer:

- Variables de control (MV/*manipulated variables*)
- Perturbaciones medibles (MD/*measured disturbances*)
- Perturbaciones no medibles (UD/*unmeasured disturbances*)
- Variables de salida/controladas (CV) medibles (MO/*measured outputs*)
- Variables de salida/controladas (CV) no medibles (UO/*unmeasured outputs*)

Puesto que la matriz de la función de transferencia tiene tantas filas como variables controladas y tantas columnas como variables de control, un sistema quedara completamente definido cuando la matriz sea cuadrada o el número de filas sea mayor que el de columnas. En este último caso el sistema estaría sobredeterminado. Entonces, una vez explicado el objetivo del control predictivo, es fácil entender que éste mismo se aplica cuando el sistema tiene ciertos grados de libertad.

Son tres los elementos que caracterizan y diferencian al control predictivo de cualquier otro:

- El modelo de predicción: a medida que se encuentra un modelo más acertado a lo que sería reacción real, las salidas son más controlables, estables y rápidas. En función del horizonte de predicción (referente al tiempo que se emplea para crear el modelo de predicción, siempre mayor al tiempo de estabilidad previsto) es más acertado o menos, y con mayor o menor carga computacional.
- La función de coste: se relaciona con todos los errores de seguimiento o en régimen permanente que pueda tener el sistema. El objetivo del control predictivo siempre será minimizarlo. Depende de las capacidades físicas del sistema, por lo que existe la posibilidad de que se le apliquen restricciones.
- Estrategia deslizante, pues sólo se aplican los controles incrementales.

El funcionamiento del control predictivo es el siguiente [18]:

1. Se predice la salida de cada instante del tiempo discreto en un horizonte de predicción determinado.
2. Se define la trayectoria de referencia que comenzará igualando referencia y salida y evolucionará hasta que se encuentre en el punto de operación.
3. Teniendo en cuenta la función de coste, con sus pesos y los errores que se estén calculando se definen los mandos de control.
4. Se toma en cuenta exclusivamente el primero de los elementos del control, pues el siguiente a la hora de aplicarlo estaría desplazado en tiempo. De manera más explicada, si el control predictivo ha calculado el control en $t+1$ mientras estaba en t , cuando esté en $t+1$ el control óptimo será el que se calcule en $t+1$, no el que se había calculado en el tiempo anterior.

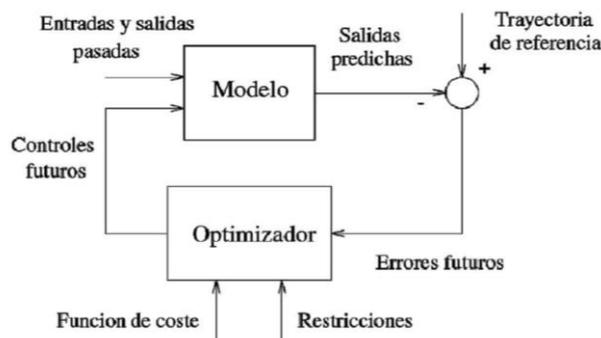


Figura 26: Estructura del funcionamiento del control MPC [18]

7.3 DISEÑO DEL CONTROL PREDICTIVO

A la hora de diseñar el control predictivo para este proyecto, se ha utilizado la extensión de Matlab del *MPC Designer*, pues es la herramienta de diseño de este tipo de controles más utilizada a nivel académico.

Lo primero de todo es diseñar la forma del sistema a diseñar, en este caso un modelo estado-espacio (*state-space*). Para ello se utiliza el modelo lineal de la Figura 13 y se trabaja con este modelo. El resultado se obtendría de la siguiente manera:

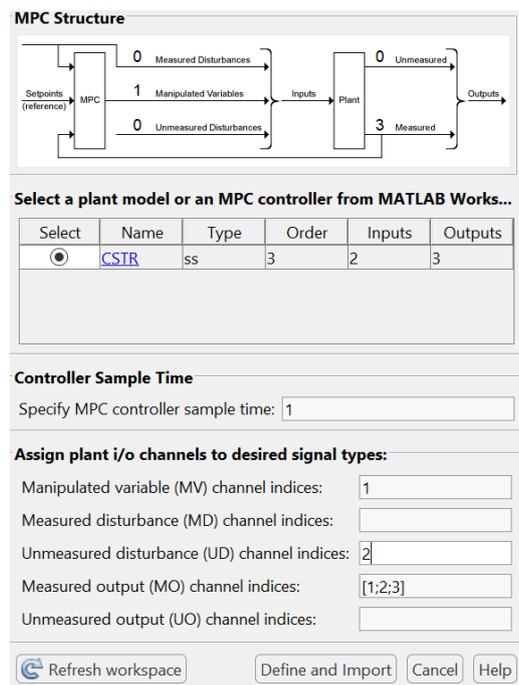
```

1. A = [-1/0.245 0 0; 1 0 0; -3.88e-3 -0.5 0];
2. B = [1/0.245 0; 0 -1; 0 0];
3. C = [1 0 0; 0 1 0; 0 0 1];
4. D = zeros(3,2);
5.
6. CSTR = ss(A, B, C, D);

```

Por tanto, ya se puede empezar a modificar la forma de las variables y la declaración de cada una de ellas. Puesto que se quiere buscar una mejora respecto al control PID, se decide trabajar con una variable de *unmeasured perturbation* que sea el giro de la pared, de tal manera que el control predictivo tenga que trabajar con esta variable desconocida.

A la hora de proceder, lo primero es añadir el diseño al programa, declarando también las diferentes variables de la manera correcta:



MPC Structure

Select a plant model or an MPC controller from MATLAB Works...

Select	Name	Type	Order	Inputs	Outputs
<input checked="" type="radio"/>	CSTR	ss	3	2	3

Controller Sample Time
Specify MPC controller sample time: 1

Assign plant i/o channels to desired signal types:

Manipulated variable (MV) channel indices: 1

Measured disturbance (MD) channel indices:

Unmeasured disturbance (UD) channel indices: 2

Measured output (MO) channel indices: [1;2;3]

Unmeasured output (UO) channel indices:

Refresh workspace Define and Import Cancel Help

Figura 27: Definición de las variables de entrada y de salida

A continuación, se procede a diseñar de manera que el salto en la salida de distancia a la pared no parta desde cero, pues los valores obtenidos no serían realistas. Más tarde, se debe definir un horizonte de control y uno de predicción, al igual que un periodo de muestreo, que no es necesario que sea excesivamente pequeño.

Puesto que lo que más se quiere medir a la salida y se quiere controlar es la distancia a la pared, a la hora de modificar los pesos, se dará todo el peso a la salida 3, que es la referente a esta variable.

A la hora de crear el escenario de simulación, se procede a aplicar un escalón en la referencia de distancia al escalón y un pulso en la UD que sea equivalente al giro que tendría que hacer de no saber que existe por ahí. Lo sucedido es lo que aparece en la Figura 28:

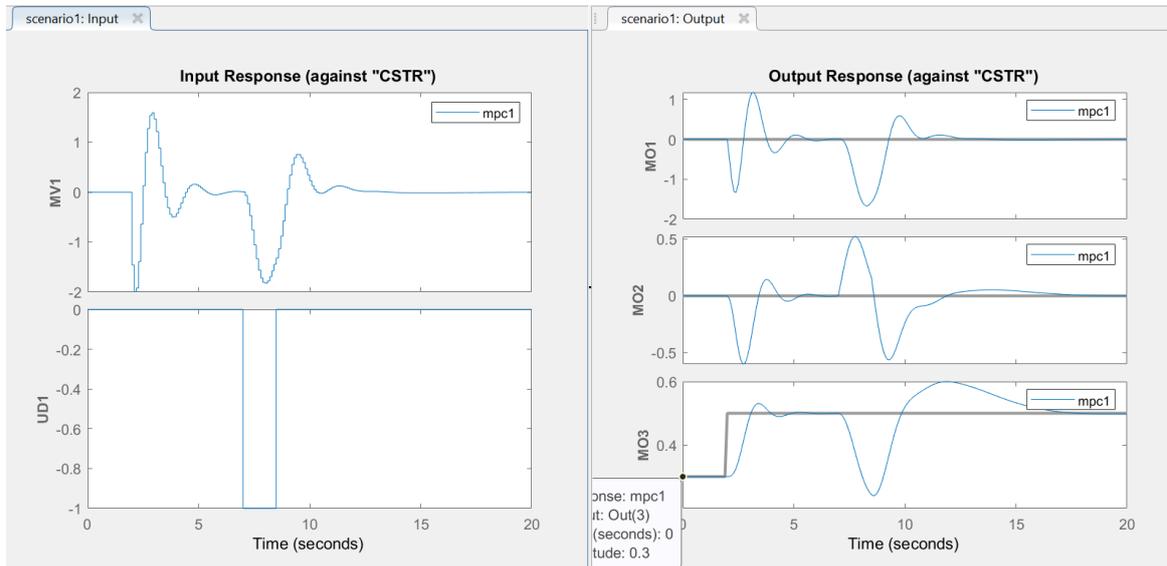


Figura 28: Gráfica de entrada y salida del control predictivo

Capítulo 8. ANÁLISIS DE RESULTADOS

A la vista de los experimentos realizados, se procede al análisis de los resultados obtenidos con el control PID.

8.1 RESULTADOS OBTENIDOS CON EL CONTROL PID

Una vez realizados numerosos experimentos con cada una de las plantas se decide elegir el mejor control para cada etapa. Comenzando por el control para planta de velocidad de avance, siguiendo por el control para planta de velocidad de giro y finalizando el análisis con el control para planta de seguimiento a pared.

Los ensayos realizados para averiguar cuál sería el control óptimo para la planta de velocidad de avance se resumen en tres: control P, control PI y control PID. En la Figura 29 se representan en una misma gráfica los resultados obtenidos tras el ensayo realizado para cada uno de los algoritmos de control. Aunque se refleja cierta similitud, especialmente al alcanzar la estabilidad, a simple vista se puede apreciar una mayor estabilidad inicial con el control PI, pues realiza menos saltos una vez alcanzada la velocidad buscada, que en este caso es 0.5 m/s. Al no ser especialmente clara la gráfica, se decide realizar una media ponderada de los valores de la velocidad de avance en cada uso de los controles, con el objetivo de encontrar el control con velocidad media de salida lo más cercana a 0.5 m/s. Los resultados obtenidos se representan en la Tabla 14.

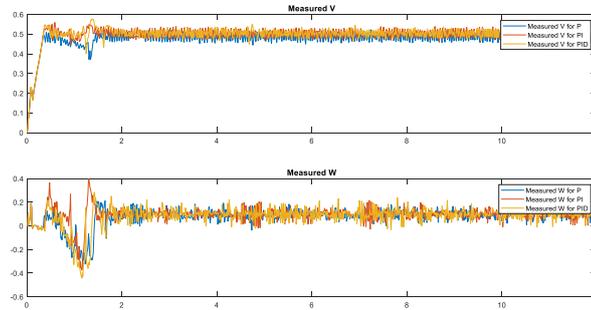


Figura 29: Representación de los controles clásicos para plantas de velocidad de avance y giro

	Valor medio de v	Valor medio de w
Control P	0.47769	0.074544
Control PI	0.4974	0.093312
Control PID	0.4927	0.081123

Tabla 14: Valores medios según el control utilizado para planta de velocidad de avance y giro

Como se observa en los resultados, tanto de la gráfica como de la tabla, el valor más correcto y cercano a la referencia es el obtenido con el control PI, sin ser descartable completamente el control PID, pues le sigue muy de cerca.

Centrando ahora el análisis sobre los controles empleados para la planta de velocidad de giro, se procede de la misma manera. En esta ocasión la gráfica es un poco menos clara que la anterior. También se puede observar que el control es un poco más lento que el de la planta de velocidad de avance. Respecto al control óptimo para gestionar esta planta, no se encuentra aparentemente uno que descarte a los otros en función de la gráfica, pues todos resultan un poco bruscos, especialmente durante los dos primeros segundos. Una vez se han establecido las señales, tampoco se observa una diferencia aparente. Procediendo a continuación al análisis de los valores medios de velocidad de giro, se observa una clara diferencia en el control PI, puesto que la prueba había sido realizada para un giro de 0.1 rad/s.

Como respuesta al análisis final, se decide que los controles óptimos para la navegación libre son los controles PI, pues consiguen reducir el error de seguimiento sin ralentizar en exceso el tiempo de estabilización de la salida.

Respecto a los controles estudiados para el seguimiento a pared, sólo se consiguió el funcionamiento de uno de ellos. El tipo de control escogido para esta planta es un control en cascada, que tendría un control aplicado a la planta que tiene como salida el ángulo del robot con la pared y otro que controla toda la planta, que tiene como salida la distancia a la pared. Puesto que el robot no se detiene para posicionarse con respecto a la pared, es primordial escoger el ángulo que va a mantener con esta para separarse, por lo que se valora la velocidad utilizada en esta planta. Es por esto, que un control PD en el lazo interno ayuda a que se estabilice lo antes posible y que no choque con la pared. El hecho de añadir un control proporcional en el lazo interno, como se pretendía con el segundo experimento, no aporta la velocidad requerida para indicar al robot la orientación correcta para que éste pueda separarse. El resultado obtenido al realizar el ensayo con el control en cascada con control PD en el lazo interno y P en el externo es el de la Figura 30:

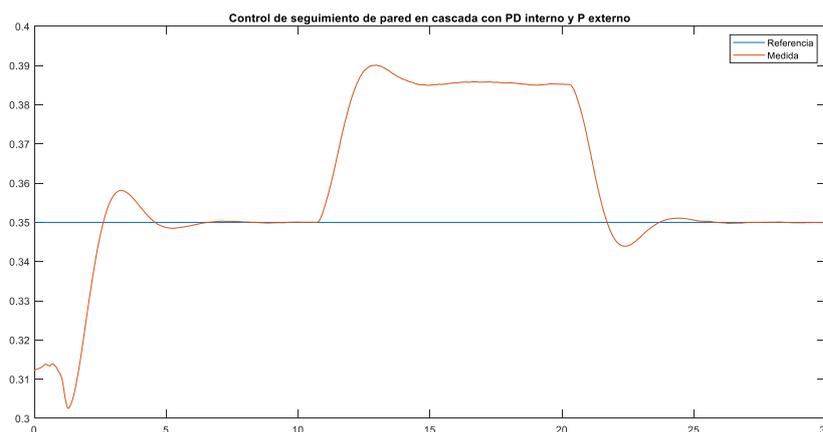


Figura 30: Gráfica de la distancia del robot con la pared con el control en cascada en simulación

Se debe señalar, que el robot se desplaza ligeramente de la pared y deja de seguir la referencia al encontrarse con un giro, pero siempre entrando en una estabilidad cuando está circulando y siguiendo la pared en giro. Los parámetros resultantes de este control en cascada son:

K_{PD}	-4.0816
T_d	0.245
b	1
N	10
K_{ext}	3.24

Tabla 15: Parámetros del control final

Como objetivo final, se ensayó el robot en un circuito cerrado con la forma de la Figura 31. Se decidió hacer dos pruebas para buscar la solución óptima. La primera opción fue emplear el conjunto de todos los controles que se habían diseñado, combinando así los de navegación libre con el de navegación mediante seguimiento de pared. Puesto que el de velocidad de avance era demasiado rápido en comparación a los otros dos, se tomó la decisión de fijar una velocidad constante en 0.5 m/s y trabajar exclusivamente con el control de seguimiento a la pared. Tras pequeñas modificaciones del control, se consiguió que éste completase una vuelta en el tiempo de 1 minuto y 12 segundos.

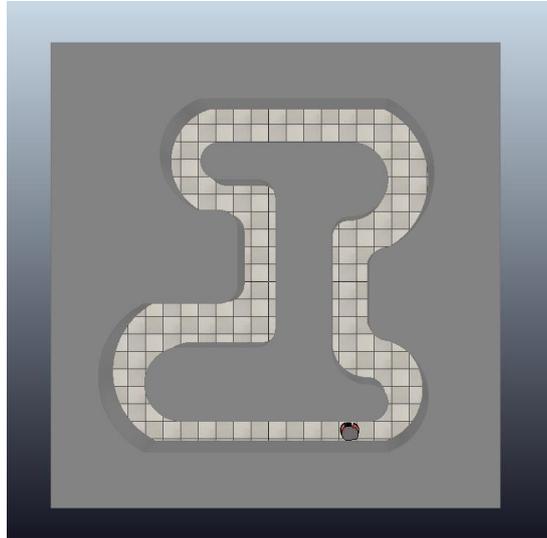


Figura 31: Entorno de Vrep para la simulación final del control

Capítulo 9. CONCLUSIONES Y TRABAJOS FUTUROS

Con el proyecto que ha sido realizado y a la vista de los resultados obtenidos, no se puede decidir entre un control predictivo o un control PID, pues no se han obtenido resultados del primero. Pero es fácil advertir la dificultad del diseño de ambos controles, especialmente la del control predictivo.

Una vez se ha estudiado y entendido el funcionamiento de este control más moderno, se razona que a la vista de los resultados obtenidos con el control PID, no sería necesario la investigación de cómo sería el funcionamiento con el predictivo, pues necesitaría una carga computacional mayor. Además, en el proyecto se ha supuesto que la medida del giro de la pared es no medible, pues no se están teniendo en cuenta los 16 sensores, pero a la hora de trabajar con todos ellos, sería perfectamente medible prácticamente todo el entorno que rodea al Pioneer-3DX.

En cuanto a los objetivos cumplidos, se ha conseguido que el robot se desplace a una velocidad de 0.5 m/s siguiendo un circuito cerrado.

En cuanto al problema del control predictivo, puesto que este es un proyecto de fin académico que no ha sido finalizado, sería posible retomarlo y plantear de nuevo el problema encontrando un resultado para este control, posiblemente buscando otro factor no medible que pudiese condicionar al vehículo y garantizar una mejora del desplazamiento gracias a la consideración de este nuevo parámetro.

Capítulo 10. BIBLIOGRAFÍA

- [1] del Val Román, J. L. (2016). Industria 4.0: la transformación digital de la industria. In Valencia: Conferencia de Directores y Decanos de Ingeniería Informática, Informes CODDII. <http://coddii.org/wp-content/uploads/2016/10/Informe-CODDII-Industria-4.0.pdf> (Último acceso: 05/06/2020)
- [2] Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A review on internet of things (IoT). *International Journal of Computer Applications*, 113(1), 1-7. <https://www.semanticscholar.org/paper/A-Review-on-Internet-of-Things-%28IoT%29-Farooq-Waseem/af2ce7f25e9d1f3e7ffa5dbbbdd5070c611c7a19> (Último acceso: 19/08/2020)
- [3] Tascón, M. (2013). Introducción: Big data. Pasado, presente y futuro. *Telos: Cuadernos de comunicación e innovación*, (95), 47-50. <https://dialnet.unirioja.es/servlet/articulo?codigo=4423775> (Último acceso: 19/08/2020)
- [4] Mejía, O. (2011). Computación en la nube. *ContactoS*, 80, 45-52. <http://www2.izt.uam.mx/newpage/contactos/anterior/n80ne/nube.pdf> (Último acceso: 19/08/2020)
- [5] Li, J. T., & Liu, H. J. (2016). Design optimization of amazon robotics. *Automation, Control and Intelligent Systems*, 4(2), 48-52. <http://www.sciencepublishinggroup.com/journal/paperinfo?journalid=134&doi=10.11648/j.acis.20160402.17> (Último acceso: 19/08/2020)
- [6] Documento explicativo funciones API remotas con Matlab, Coppelia Robotics. <https://www.coppeliarobotics.com/helpFiles/en/remoteApiFunctionsMatlab.htm> (Último acceso 13/08/2020)
- [7] Wen, S., Othman, K. M., Rad, A. B., Zhang, Y., & Zhao, Y. (2014, January). Indoor SLAM using laser and camera with closed-loop controller for NAO humanoid robot. In *Abstract and Applied Analysis* (Vol. 2014). Hindawi. <http://downloads.hindawi.com/journals/aaa/2014/513175.pdf> (Último acceso: 19/08/2020)
- [8] Andrango, L., & Sotomayor, N. (1999). Simulación y Ensamblaje de un Prototipo para Control y Navegación de un Robot Móvil. *Jornadas de ingeniería eléctrica y electrónica*

- (18, 2000, Quito, Ecuador)
- https://www.researchgate.net/profile/Nelson_Sotomayor/publication/27557540_Simulacion_y_ensamblaje_de_un_prototipo_para_control_y_navegacion_de_un_robot_movil/links/5406062d0cf2bba34c1e3b6c/Simulacion-y-ensamblaje-de-un-prototipo-para-control-y-navegacion-de-un-robot-movil.pdf (Último acceso: 19/08/2020)
- [9] Poudel, D. B. (2013). Coordinating hundreds of cooperative, autonomous robots in a warehouse. Jan, 27, 1-13.
- <https://www.semanticscholar.org/paper/Coordinating-Hundreds-of-Cooperative%2C-Autonomous-in-Wurman-D'Andrea/fd40544505ff2423d7ad35f668f7031c98b0ed55> (Último acceso: 19/08/2020)
- [10] D'Andrea, R. (2012). Guest editorial: *A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead*. IEEE Transactions on Automation Science and Engineering, 9(4), 638-639.
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6295687> (Último acceso: 19/08/2020)
- [11] Así son los robots eléctricos y autónomos de SEAT Martorell que han jubilado a los camiones para el transporte de piezas, Motorpasion, David Galan (2020)
- <https://www.motorpasion.com/seat/asi-robots-electricos-autonomos-seat-martorell-que-han-jubilado-a-camiones-para-transporte-piezas> (Último acceso: 19/08/2020)
- [12] H. Batti, C. B. Jabeur and H. Seddik, "Fuzzy Logic Controller for Autonomous Mobile Robot Navigation," *2019 International Conference on Control, Automation and Diagnosis (ICCAD)*, Grenoble, France, 2019, pp. 1-6, doi: 10.1109/ICCAD46983.2019.9037922.
- <https://ieeexplore.ieee.org/document/9037922> (Último acceso: 25/08/2020)
- [13] Hernández-Arroyo, E., Díaz-Rodríguez, J. L., & Pinzón-Ardila, O. (2014). Estudio del comportamiento de un Control MPC [Control Predictivo Basado en el Modelo] comparado con un Control PID en una Planta de Temperatura. *Revista Facultad de Ingeniería*, 23(37), 45-54. <http://www.scielo.org.co/pdf/rfing/v23n37/v23n37a05.pdf> (Último acceso: 19/08/2020)
- [14] Laboratorio de Control Avanzado, ICAI, curso 2019-2020
- [15] Moreno, M. A. (2001). apuntes de control PID. La Paz Enero, 8, 6-7.
- <https://es.scribd.com/doc/23817781/Apuntes-de-Control-Pid> (Último acceso: 19/08/2020)
- [16] Regulación Automática, F. Luis Pagola, (2006)

- [17] Apuntes regulación automática, Control predictivo basado en modelo, Juan L. Zamora, Ramón Rodríguez Pecharromán y Aurelio García Cerrada, (2017)
- [18] ¿Qué es el control predictivo y hacia dónde se proyecta?, Sendoya Diego Fernando, (2012)
<http://oaji.net/articles/2017/5082-1501183303.pdf> (Último acceso: 18/08/2020)
- [19] GIL, Carlos Gómez. Objetivos de Desarrollo Sostenible (ODS): una revisión crítica. *Papeles de relaciones ecosociales y cambio global*, (140), 2018, p. 107-118.
https://www.fuhem.es/media/cdv/file/biblioteca/revista_papeles/140/ODS-revision-critica-C.Gomez.pdf (Último acceso: 25/08/2020)

ANEXO I: FREECAD CON V-REP

En el proyecto, el programa de FreeCad ha sido utilizado para el diseño de los entornos de simulación de V-REP. Al igual que con cualquier otra herramienta de diseño en 3D, lo primero de todo ha sido acotar completamente la figura a diseñar en un plano y más adelante convertirla en un *mesh*.

Una vez haya sido diseñada la figura en 3D, debe ser exportada en un formato que sea válido para V-REP, como puede ser en formato *obj*. A la hora de importar este sólido en V-REP, se debe seleccionar *File->Import->Mesh...* y escoger el entorno con el que se decida trabajar. Probablemente sea necesario reescalar la imagen a conciencia para que encaje en el entorno de simulación.

Una vez se haya encontrado la manera adecuada de adaptar el *mesh* al entorno, se debe configurar este mismo para que tenga las características físicas que se requieran, siendo la más importante de ellas la de que sea un material sólido. Para modificar estas características, se accede *Tools->Scene object properties*.

ANEXO II: ALINEACIÓN DEL PROYECTO CON LOS ODS

“Los Objetivos de Desarrollo Sostenible (ODS) adoptados por las Naciones Unidas en 2015 contienen la agenda global más ambiciosa aprobada por la comunidad internacional para movilizar la acción colectiva en torno a objetivos comunes. Si bien se proponen luchar contra la pobreza extrema, integran y equilibran tres dimensiones esenciales del desarrollo sostenible como son la económica, la social y la ambiental, proporcionando una valiosa hoja de ruta para articular la formulación de políticas mundiales.”[19]

Esta lista de objetivos está compuesta por 17 objetivos, entre los cuales muchos de ellos se caracterizan por estar relacionados con la ingeniería, en el sentido de que ésta es muy necesaria para poder cumplir los objetivos propuestos.

Entre estos objetivos y en relación al proyecto realizado, cabe destacar el ODS 7 u objetivo de energía sostenible y no contaminante, pues a pesar de que éste sea un proyecto académico, el futuro desarrollo de este tipo de proyecto llevará a cabo un tipo de control que reducirá los gastos energéticos y aprovechará al máximo los recursos.

Por el mismo razonamiento que en el anterior, este proyecto también sigue el ODS 9, de industria e innovación, el ODS 11, de ciudades sostenibles y el ODS 12 de producción y consumo responsable.